BB

DESIGN & DATA ACQUISITION at 3-D FORCE SENSOR

Saïd BOUKERMA

Université de Technologie de Compiegne - FRANCE

February - June 1989
WPA-rapport nr.0724

UNIVERSITY OF TECHNOLOGY EINDHOVEN
Faculty of Mechanical Engineering
Group WPA

1 februari 1989

Research project : S. Boukerma (Université de Compiègne)

Professor : Prof.dr.ir. J.E. Rooda

Coach : Ir. P.C. Mulders
: Ing. H.A.W.M. van Rooy

Subject : Design and implementation of a 3D-force sensor.

General view

In the laboratory of control technology is built a RT-(rotation/translation) robot with a force sensor. This robot should be completed with an optimal flexible controller.
With the aid of a force sensor teach- and replay operations are studied. In spite of place- and time dependent parameters the optimal replay of trajectories must be performed.

Project

Analyse the performance of a 3D-force sensor for the teaching of a rotation-translation robot. Design the interface between the sensor and a single board computer. Construct the hard- and software with an interface-time as short as possible and test it.

Prof.dr.ir. J.E. Rooda

Ir. P.C. Mulders

ACKNOWLEDGEMENTS

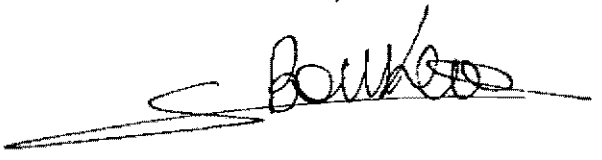
I want to express my gratitude to my coach Mr P.C. MULDERS for his kindness and efficient advice.

The organization of my project by the international relations staff was satisfying.

The nice atmosphere created in the laboratory was very pleasant thanks to all the students.

I also want to thank the students of I.R.C.E (International Reception Committee of Eindhoven) who involved themselves in the good march of my stay.

EINDHOVEN, June 1989



saïd BOUKERMA

CONTENTS

Page

I INTRODUCTION 1

I.1 The Engineering Education in the Netherlands .. 1
I.2 Eindhoven University of Technology 1
I.2.1 The Mechanical Department 2
I.3 My Practical Work 3

II DIFFERENT TEACHING METHODS 4

II.1 Method used 4
II.2 Other Teaching Methods 4

III THE R-T ROBOT 6

III.1 The Mechanical Part 6
III.2 The Actuators 6
III.3 The Interfaces 8
III.3.1 The Force Sensor Interface 8
III.3.1.1 General Information 8
III.3.1.1.a Justification of this Interface 8
III.3.1.1.b Review of the Previous Interface ... 8
III.3.1.1.c New Ideas 9
III.3.1.1.d Other Solutions 11
III.3.1.2 Technical Information 11
III.3.1.2.a The SDM 856 11
III.3.1.2.b The Filter 12
III.3.1.2.c Programmable Amplifier: AD 526 13
III.3.1.2.d Offset Adjustment 14
III.3.1.2.e Overrange Detection 14
III.3.1.2.f Channel Selection 15
III.3.1.2.g Interface Timing 15
III.3.1.2.h Acquisition Speed 15
III.4 The Interface iSBC/Motor Power Amplifier 16
III.5 The Position Interface 16
III.6 The Interfaced Devices 17
III.6.1 The Force sensor 18
III.6.1.1 Mechanical Consideration 18
III.6.1.2 The Strain Gages 20
III.6.1.3 Electronic Description 21
III.6.2 The Hall Switches 21
III.6.3 The Position Force Sensor 21
III.6.4 The Power Servo Amplifier 22

IV THE MULTIPROCESSOR SYSTEM 23

IV.1 The Communication 23
IV.2 Multibus System 24
IV.2.1 the Multibus Arbitration 25
IV.3 The RAM Memory Board: iSBC 028A 26
IV.4 Single Board Computers 26
IV.4.1 iSBC 186/03 26
IV.4.2 iSBC 86/12 27
IV.5 Memory Addressing Scheme 29

V THE SOFTWARE 30

V.1 Program Development 30
V.2 Program Structure 30
V.3 Calculations 32
V.4 Interrupt Procedures 33
V.5 Definition of the Various Modes 34
V.5.1 Test Mode 34
V.5.2 Replay Mode 34
V.5.3 Teach Mode 34
V.6 Peripherals 34
V.6.1 PPI 34
V.6.2 PIT 35
V.6.3 PIC 36
V.7 Development System 37
V.7.1 Presentation 37
V.7.2 Compiling and Linking Operations 38

VI CONCLUSION 39

VII APPENDICES 40

I INTRODUCTION

I.1 The Engineering Education in the Netherlands

Engineering education at University level in the Netherlands is not surprising, because the engineering education is not based like in France on two structures; schools and universities.

In the Netherlands, there are three universities which like the TUE (Technische Universiteit Eindhoven) award the engineering degree. The TUE was founded in 1956, the oldest located in Delft in 1842 and the most recent one located in Twente in 1961.

Technical Universities have a total population of 20,000 students.

The first degree which concludes a course of studies is the master degree. In engineering subjects, the master degree entitles the graduate to use the legally protected abbreviation `Ir' (Engineer) before his name. To get the Doctor degree 'Dr' no specific courses are required. The student has to be 'Ir' and is required to carry out original research.

I.2 Eindhoven University of Technology

This university offers nine courses of study which students can pursue.Engineering titles are granted in the following fields:

- Technology in its social application (60 students)
- Industrial engineering and science management (1150 students)
- Mathematics (360 students)
- Computer Science (438 students)
- Technical physics (529 students)
- Mechanical engineering (872 students)
- Electrical engineering (1083 students)
- Chemical engineering (656 students)
- Architecture, Structural engineering and urban planning (716 students)

Among these students, 7% are female. The academic and administrative staff is composed of 2200 persons who have permitted the delivery of about 9000 diplomas since the opening of the faculty.

I.2.1 The Mechanical Department

This department is the second largest of the university, subdivided in many sections, Design and Production are the most significant sections in which the diverse field of mechanics including scientific research and development and industrial application are represented.

The courses are focused on eight topics, reflecting the major sections:

- Control and simulation of mechanical manufacturing processes
- Biomedical technology of vital human biological functions
- Design of process equipment for maximum reduction of life cycle costs
- Non linear dynamics and random vibrations
- High efficiency power transmission and tribology
- Optimization and energy conservation in transport
- Design of machines for making reinforced plastic materials
- Flexible automation and industrial robots

I worked during 5 month in the WPA section (engineering and automation production) and participated to the FL.A.I.R.project (FLexible Automation and Industrial Robots). The research project flair is financed and directed by the Dutch government. Its goal is to get some experience in flexible automation and industrial robotics.

In this project each university has its own task. At the TUE this program involves both the mechanical and the electrical department and including relations with private companies.

The programm is divided into five sections:

- The general aspect of automation
- The handling of parts
- Kinematics and Dynamics of mechanical structures
- The drive and control systems and their applications
- The arc welding and the sensory system

I.3 My Practical Work

Working on the sensory system which is a part of the F.L.A.I.R project, my coach ir P.C. Mulders is developing a two dimensional robot. This robot has been built in many steps, with first only one degree of freedom; it is now going to get a second one.

The original feature of this robot is the fact that it includes a force sensor which is used to calculate the forces and moments applied on the robot.

During my project, I was busy with this force sensor. We had to establish a link to the computer,which was Analog/digital interface.

An interface which has been previously built didn't provide sufficient speed and accuracy.

An interface, of course must be connected to computer. therefore, I also worked with an INTEL single board iSBC 86/12A in order to develop the software.

The organization of my work was as follows; at the beginning I was to get some information on the robot, the control, the parallel processing, the Intel boards, the former student's work and finally to precisely define the field of my work. After that, my task was to design a new interface eliminated the previous design efficiencies. This was done in cooperation with my coach and Mr Henk van Rooy, who helped me a lot in this crucial part of my project.

Once the design was finished, we didn't have enough time to build the interface. We let the CTD (technical service of the university) build it.

At that point, the software had to be written. Before starting it, we had to choose the language. The assembler which appeared best adapted to that application seemed too complex to be mastered, so the pascal was used.

The last part of the work was the coordination of the software and the hardware, usually the most difficult task.

Unfortunately, the building of the interface required much more time than expected. So,the interface was not completely tested. This will be the first step of another project.

II DIFFERENT TEACHING METHODS

The purpose of this robot is to improve one of the numerous teaching techniques.
A teach method is a more or less original way to record a trajectory with the expectation that the robot will copy this trajectory.

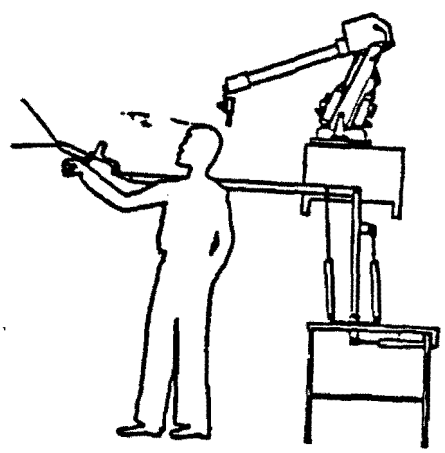
II.1 Method Used

A recently developed method integrates a six-axis force/torque sensor in the robot. This method allows immediate programming of robots either for paths or simultaneously for forces and torques to be exerted on the robot's environment. It avoids all the difficulties inherent with any kind of off-line or CAD programming caused by uncertainties in the geometry of robot and in the material to be processed. This method is based on a force sensor, which can be used by the operator or directly placed at the end of the robot's arm.

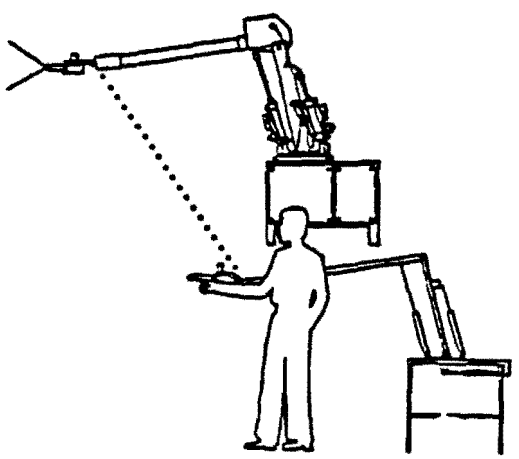
II.2 Other Teaching Methods

Among the several methods of teaching, in addition of the force sensor method, three methods are predominant:

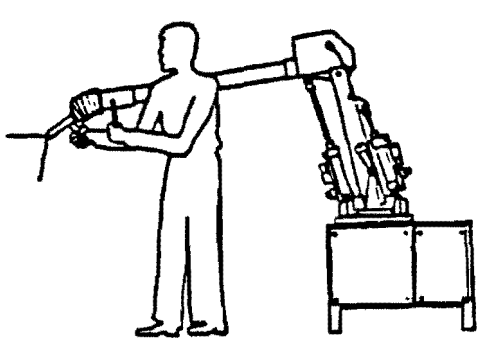
- The manual control: We use a reversible robot which is able to be manually moved and can register a trajectory. The limit of such a control is obvious. It is applicable solely for small robots and accuracy of the control is also small.
- The dummy arm method: In this method, we use a mechanical structure identical to the robot arm but not motorized and very light weight. It includes the different position and velocity sensors. As we use a different mechanical arm, the accuracy of the copied motion is not as good. This method has the disadvantage of requiring a second mechanical structure.
- The telecontrol method: The method is based on the use of a teach-pendant or a joystick. The user can observe the effect of his commands. The coordination of several degrees of freedom is impossible using a teach-pendant. Another problem with the joystick is that the human operator never masters the delay considerations.



dummy



teleconcrol



manual control

fig.1: Teaching methods

III THE R-T ROBOT

III.1 The Mechanical Part

The mechanical specifications of the robot are the following:

Translation part:

- Maximum stroke : ± 70 cm
- Maximum linear velocity : 1 m/s
- Maximum linear acceleration : 10 m/s^2
- Maximum load : 50 kg
- Linear accuracy : 0.01 mm
- power transmission : screw and spindle
- mechanical energy provided by a DC motor

Rotation part:

- Maximum rotation: π rad
- Maximum velocity: $\frac{1}{2}\pi$ rad/s
- Maximum acceleration: $\frac{1}{2}\pi$ rad/s^2
- Circular accuracy: 0.002°
- mechanical energy provided by a DC motor

III.2 The actuators

Each actuator of the robot is driven by a DC motor (MC 19 p from the Electro Mechanic Company: CEM) which requires a power amplifier unit (Axodyn Series 05 LV power amplifier from Brown Boveri Company: BBC).

This AXEM motor is fitted with a flat rotor with lamellar conductors (armature) and permanent magnet (field).

This motor can also be associated with AXEM Tacho-Generators, Brake, Gear Box and Encoder.

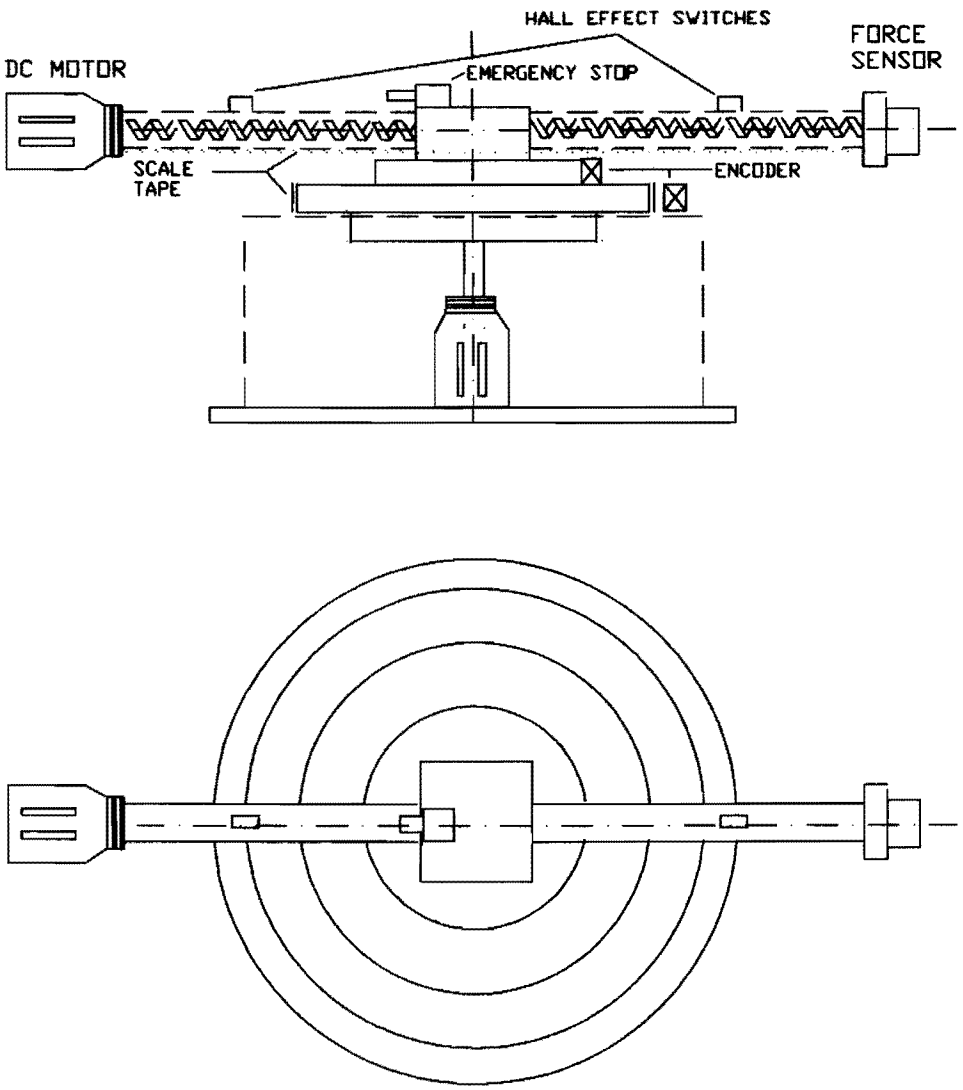


fig.2: Schematic of the Robot

III.3 The Interfaces

Allowing the robot to communicate with the multiprocessor system, the interfaces often limit the dynamic possibilities of a robot. In our case we tried to design simple but efficient interfaces.

III.3.1 The Force Sensor Interface

It is not reasonable to design something for which the use and the peculiarities are not clearly defined.
Thus, the objective of the interface must be defined as well as the drawbacks of the original one. We must also study many solutions and take the optimal one, which is justified from a technical point of view and is financially acceptable.
The second part will show details and explanations to understand how the interface works.

III.3.1.1 General Information

III.3.1.1.a Justification of this Interface

An interface is very specific and has to be designed or adapted for any need of communication between a computer and a peripheral.
In this case, we needed to convert eight analog signals coming from the force sensor (Refer Specific Chapter) to digital signals. We have to carefully consider the speed of the acquisition, the quality of the signals, the optimal use of the force sensor, the financial investment and possibly most importantly; give the interface many options in order to allow for further development.
This interface has an important place in the control of the robot. That is why, it has to be very fast, accurate and not too sensitive to noise.

III.3.1.1.b Review of the Previous Interface

The first interface designed for this two dimensional robot didn't satisfy to the requirements of the real time control.
It had no handshaking with the computer and the use of the software for generating a clock rate was ineffective. Moreover, the filter which is the guarantee of a better signal in the acquisition chain, wasn't implemented.

Nevertheless we have also to consider the good points of this interface. The use of the SDM 856 was a good idea (See in Appendix the technical data sheets and below the technical information), the adjustment of the offset is correct and the decoupling between digital and analog ground is effective.

III.3.1.1.c New Ideas

We kept the SDM 856 component, because it solves many problems and avoids the use of three chips (Multiplexer, Sample/ Hold, Analog/Digital Converter). But instead of using an arbitrary acquisition directed by the computer, we decided to use the control lines of the SDM 856, the interrupt lines and the timer of the computer. In this manner, a lot of time is saved.

The use of a filter appeared obvious after the checking of the analog signal shape.

Instead of amplifying the analog signal between the multiplexer and the sample/hold it seemed more logical to increase the amplification range of the amplifiers in the force sensor. Thus, the settling time of the converter components is reduced and despite the longer settling time of the sensor amplifiers, we get a faster acquisition.

Previously if an overrange occurred one LED was lit, but no information went into the computer. We use now an interrupt line (Overrange interrupt) to inform the CPU of such events.

The former acquisition needed two steps. The twelve bits from the DAC were connected to eight lines of the PPI, which meant that we had to enable and disable the output latches of the DAC. The output of the DAC's latches are in the new board, individually connected to an input line of the PPI.

In the first design, the selection of the channel was done by the CPU through the PPI. The new card owns a counter which performs this task.

An optimal use of the force sensor is realised thanks to a programmable amplifier. In this way, there is the opportunity to use the force sensor as a joystick and manually applied constraints.

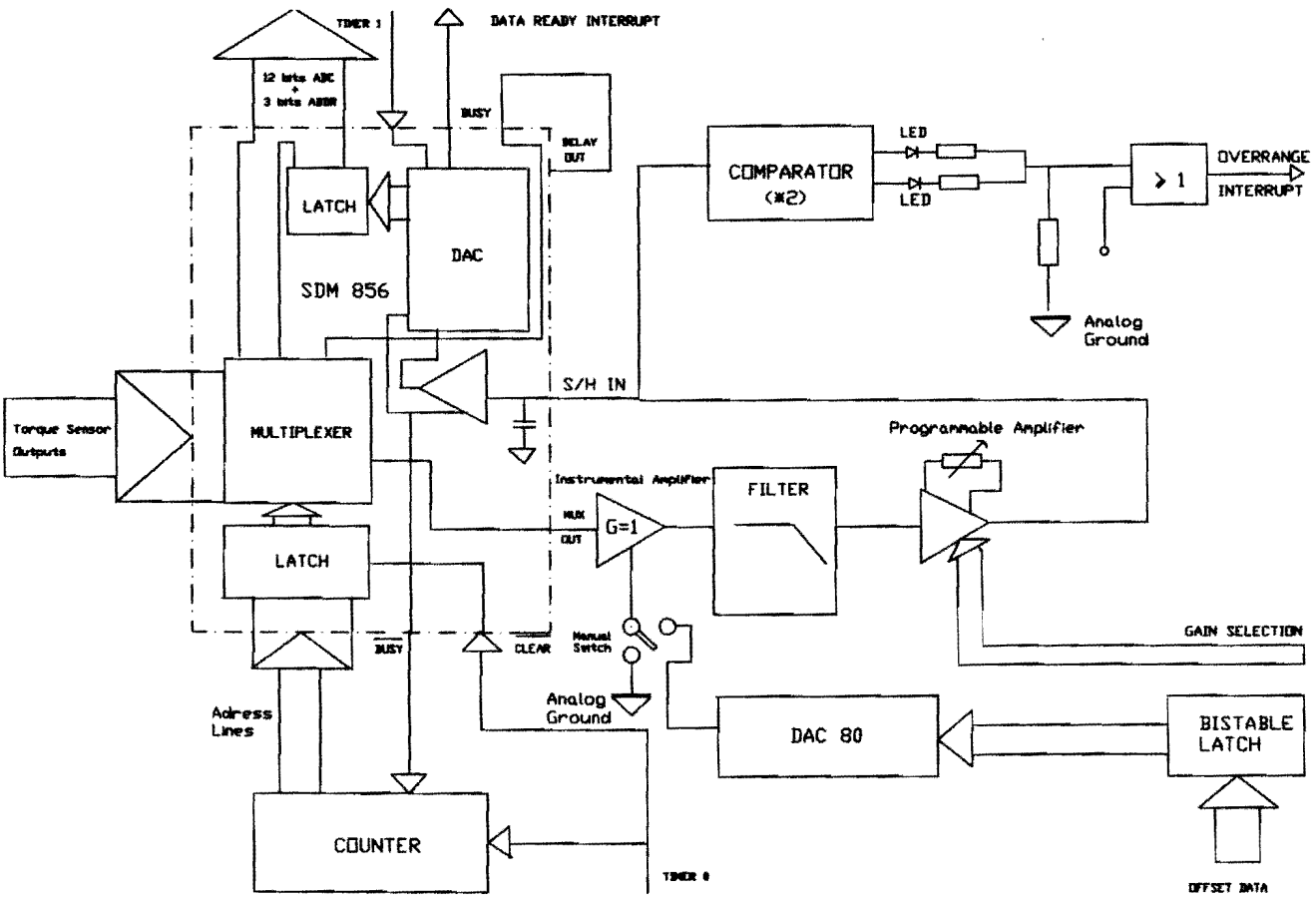


FIG.3: SCHEMATIC of the INTERFACE

III.3.1.1.d Other Solutions

A more logical solution might be to use eight filters and eight sample/hold units. in this case, the eight signals are recorded at the same time. For economic reasons, this solution has been ruled out.

Another solution is to use the PING-PONG acquisition which is a specific way to run two Sample/Hold units in parallel. Thus, we can eliminate the sampling time; because when one S/H is in the Hold state, the other one is in the Sample state.

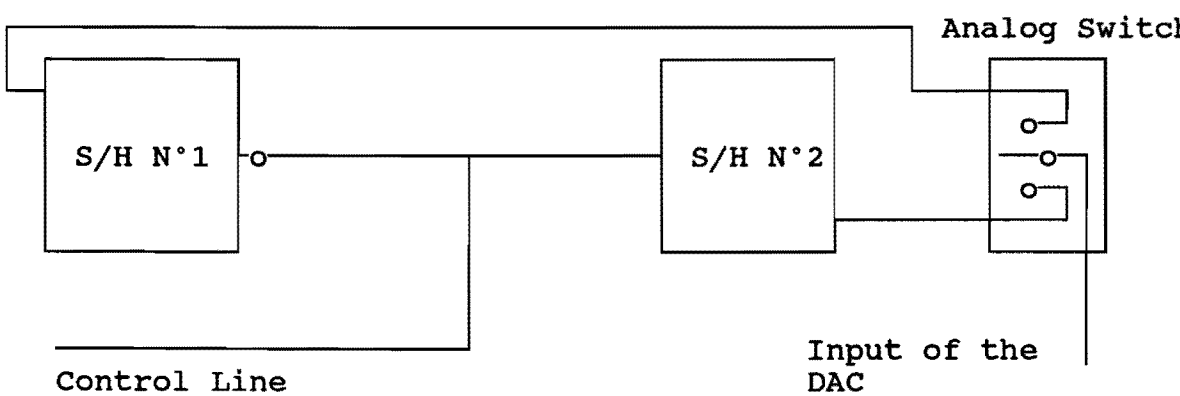


fig.4: The PING-PONG technique

III.3.1.2 Technical Information

III.3.1.2.a The SDM 856

This component, sold by Burr-Brown is the skeleton of the interface. It includes a multiplexer with eight differential inputs or sixteen single-ended inputs and latched addresses, a Sample/Hold which is also rather slow (Acquisition time: 10 μs) and an ADC (Analog Digital Converter) not very quick neither (Conversion time: 30 μs). The integration of many components into one brings advantages, such as reliability, smaller size, less noise. However, it is more fragile.

We use it in a special mode called overlap mode, that is to say the selection of the next channel is done during the conversion of the presently selected channel. The settling time effect of the multiplexer, the filter and the programmable gain is no longer included in the conversion time.

A delay and a clock are also available in this component, we only use the delay to synchronize the signals.

The control of multiplexer's address is done with two lines LOAD and CLEAR; the rising edge of LOAD put the data present on the address lines into the latch and CLEAR selects the channel 0.

The ADC is controlled trough the TRIG line which initiates the A/D conversion. It runs the Sample/Hold with the BUSY line.

The delay is triggered by the rising edge of the STROBE input.

III.3.1.2.b The Filter

Also from Burr-Brown the UAF 21 is an universal active filter therefore, it is adaptable to many uses:

- Low Pass
- High Pass
- Bandpass
- band reject

For each function, three configurations are allowable:

- Butterworth
- Bessel
- Chebyschev

And in any case we can choose the tuneable frequency, the gain and the Q-factor.

The UAF 21 is a two-poles active filter which requires only four resistances to be completely defined. In our application we use it as a second order low pass filter with a bandwith of 3.5 kHZ and the Bessel configuration.

By using the equations B (Refer to the specific data sheet) we calculate the various resistors:

- RF1=RF2= 11.3 kΩ
- RQ= 10.9 MΩ
- RG= 200 KΩ

With those equations, we must add a resistor of 11 kΩ between pins 12 and 1.

The use of a lowpass filter is justified by the fact that the speed of the acquisition is limited, so all the frequencies above the tuneable frequency are without interest and could decrease the accuracy of the Sample/Hold.

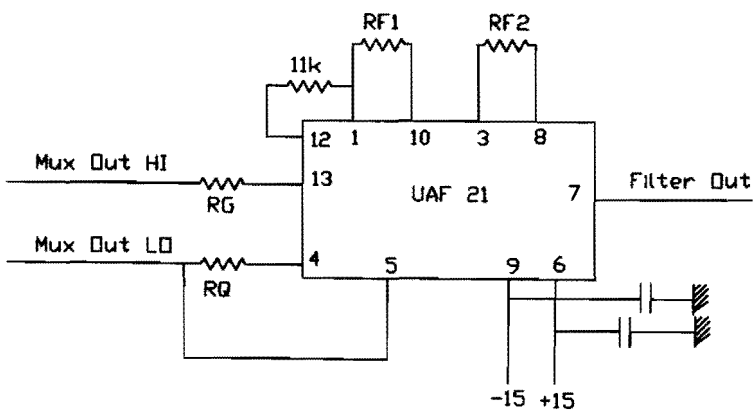


fig.5: the Filter

III.3.1.2.c Programmable Amplifier: AD 526

Supplied by Analog-Device, this component has digitally programmable binary gains from 1 to 16 and is cascadable. It provides gains of 1,2,4,8 and 16. It is complete, including amplifier, resistor network and TTL-compatible latched Inputs, and needs no external components.

We use only three ranges of amplification; 1,2 and 4. So, only the three lower bits are useful.

Two modes of operation are available; transparent and latched mode of operation. In the first one, the gain changes directly with the level changes of the gain code inputs contrary to the second mode which requires to select a control line (CS or CLOCK) to modify it.

In our application, the transparent mode is used, so the CS and CLOCK lines (Pins 14 and 13) are connected to the ground.

Using this component permits a wide use of the force sensor which can either be placed a the end of the robot arm or used as a joystick with manual control. Moreover, this component doesn't require any Digital/Analog interface and offers very good dynamic characteristics.

III.3.1.2.d Offset Adjustment

This is a debatable point, because we can consider that on one hand, this adjustment of offset doesn't bring more precision and requires settling time. On the other hand, it seems interesting to reduce the conversion time of the ADC.

It is realised with three components:

- a Digital/Analog Converter (DAC 80 from Burr-Brown)
- a Bistable Latch (SN 74100 from Texas Instrument) used to synchronize the offset adjust system with the control lines.
- an Instrumental Amplifier (INA 101 from Burr-Brown) with a gain of 1, the output of the DAC is connected to the reference signal of this component.

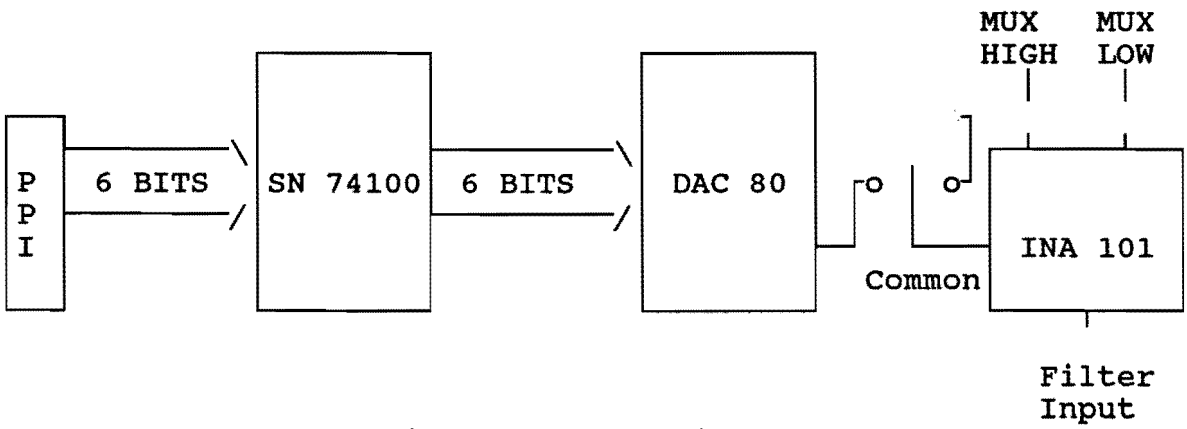


fig.6: Offset Adjustment

III.3.1.2.e Overrange Detection

Already present on the first design, this system has been adapted to provide an interrupt signal to the computer when an overrange occurs.

It is composed of one major component (LM 747D), containing two operational amplifiers used as comparators.

With a resistor network, we select the overrange level. Each output of the Op-Amps is connected to a voltage divisor which includes a LED and provides a five volt level. To make this output signal TTL-compatible we use an AND gate (SN 7432).

III.3.1.2.f Channel Selection

A four bit synchronous up/down loadable counter (SN 74193 from Texas Instrument) is used to select the channels of the multiplexer. The inputs are connected to the ground and the LOAD line is used as a CLEAR input. This counter is incremented through the BUSY line of the SDM 856 and cleared with the output of the timer 0 also connected to the CLEAR pin of the SDM 856.

The addresses of the multiplexer are registered by the computer to check the data.

III.3.1.2.g Interface Timing

In this area we allow a total flexibility, all the control lines are put on a socket and many configurations allowed.

In any configuration, all the addresses (SDM 856 and SN 74193) have to be at zero before each acquisition. The ADC is triggered with the output of the timer 1 and the Sample/Hold control line is connected to the BUSY pin of the DAC.

However, we allow many possibilities with the delay line which could be connected to the LOAD input of the SDM 856. This delay is adjustable from 3 to 300 μs. The use of the timer allows the user to select a wide range of frequencies.

III.3.1.2.h Acquisition Speed

This is one of the major reasons for the design of a new interface. For the first card, the maximum speed for the complete acquisition is about 1.5 kHZ, with the new one a rate of 3 kHZ is allowed.

We can see on the figure below the role of the sample time; the shorter, better the acquisition is.

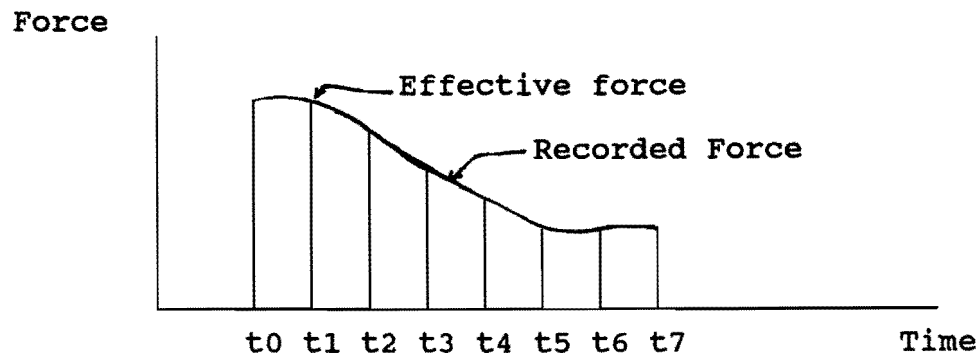


fig.7 consequence of the sampling

III.4 The interface iSBC/ Motor Power Amplifier

This interface is very simple and is composed of a twelve bit DAC (Digital Analog Converter), which is the DAC 811 from Burr-Brown which provides voltage in the range of [-8v,+8v].

In order to get the two's complemented code an inverter gate has been used for the MSB of the twelve bits issued from the iSBC.

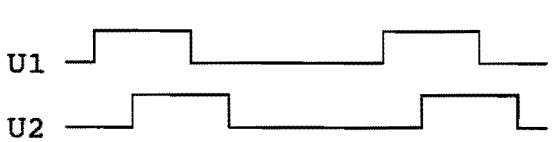
The output of the DAC is connected to an Operational Amplifier (OPA 27 from Burr-Brown), which brings the signal into the range of [-10v,+10v].

III.5 The position Interface

The position is coded with 20-bits, but the interface is designed with 16-bit (discriminator) and 4 bit (counter) components. The acquisition is achieved in two steps: sixteen and then four bits.

The signal issued from the position sensor allows sense discrimination. It consists of two pulses shifted in phase which are generated by the position sensor.

Pulses issued from the Position sensor:



The Schematic of the interface is the following:

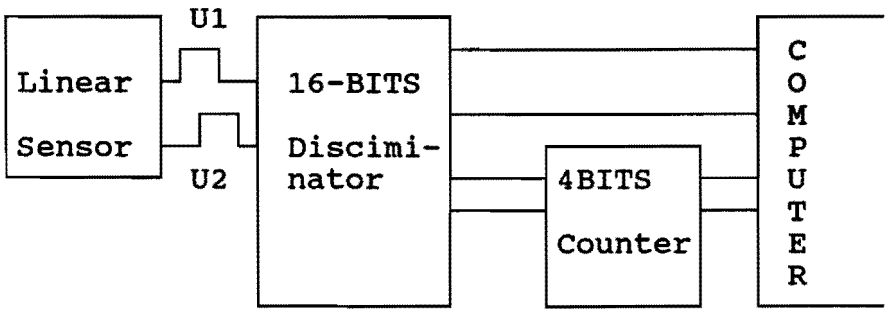


fig.8: schematic of the position interface

III.6 The Interfaced Devices

The Sensors are numerous in the control of the robot. One for the registration of the forces, two for the position recording, four for end Switches. All these sensors are interfaced to the iSBC.

To run the two motors, we also use two identical interfaces which are connected to the power electronics.

The schematic below gives some ideas about the interfaced devices.

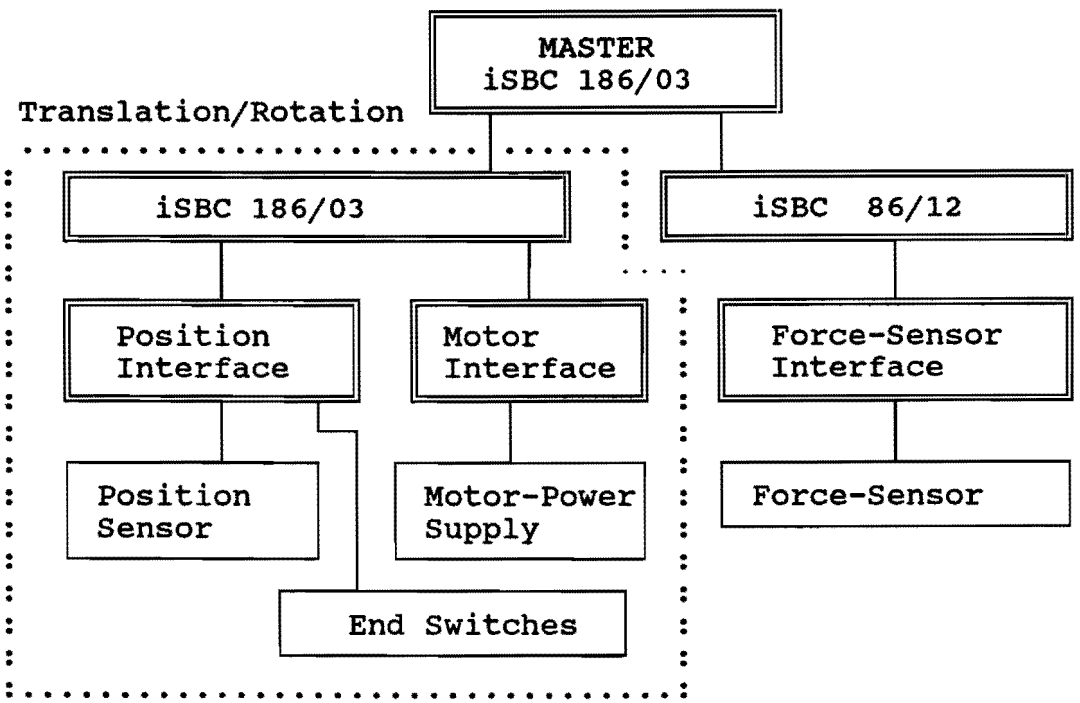


fig.9: Block Diagram of the Interfaces

III.6.1 The Force Sensor

This torque sensor is a home-made sensor. It was a part of another project and it looks like an industrial one.

The signals generated by this device have to be as "clean" as possible, because they are at a very low level (≈10 mV) and must be amplified.

The sensor description will be divided in three parts; a mechanical analysis, a strain gage analysis and an electronic analysis.

III.6.1.1 Mechanical Considerations

The structure of this three dimensional torque sensor is in aluminum. It has been built with two principle structures; a base which can be mounted on the robot arm and a cover on which the "helptool" can be fixed.

Both parts are linked together through connection bars. Six bars are needed to ensure a static position of the two parts, the use of more bars would be detrimental to the swiftness and the symmetric structure of the sensor.

The main problem with the construction of the sensor is to get the best orientation for the connection bars, because of the small volume required for the interface.

The mechanical part consists of a circular base which is connected to the nucleus by four bars, itself connected to the upper ring with four bars.

On each bar there are two strain gages as close as possible to the nucleus. Together they form a half Wheatstone bridge.

If forces or moments are performed between the two rings, they cause elastic distortions which are converted into voltage by the strain gages.

With the eight signals, the constraint on the sensor can be calculated by using a transformation matrix which links the eight voltages to the six components of the constraint matrix.

As said above, six bars are sufficient to calculate the efforts applied on the sensor. The use of eight bars permits the calculation of the same constraints, but as we can see in the transformation matrix, there are many null terms. So the calculation time is greater and the link between the two rings is hyperstatic.

Despite the fact that steel offers a better linearity in its deformation, aluminum is used because of its lower modiolus of elasticity. Furthermore it doesn't corrode and is easy to tool. Moreover, with the use of aluminum, we can get specific strain gages.

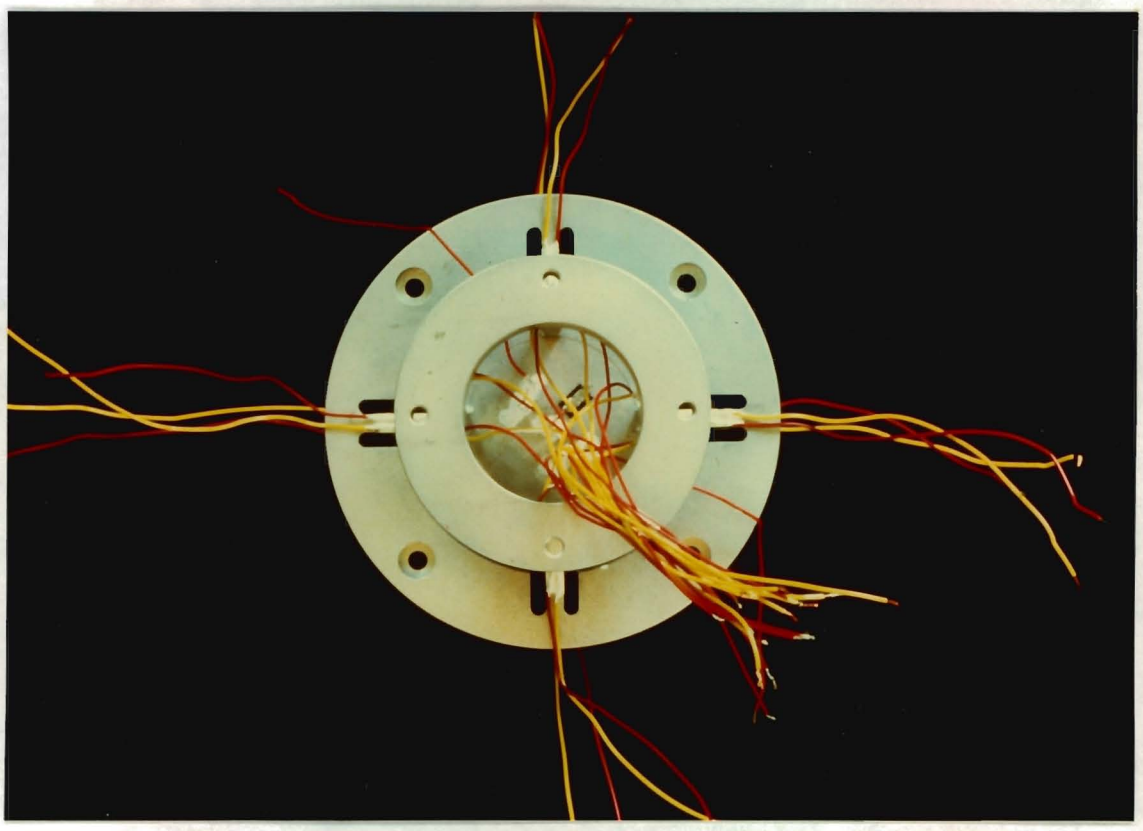


fig.10: Picture of the force sensor

The mechanical structure has been made of one piece to avoid non-linearities as a cause of hysterisis or friction with screws or non electrical connections with stick.

III.6.1.2 The Strain Gages

The principle of the strain gages is simple; they are resistors in which the relative change in resistance ($\delta R/R$) easily measured is proportional to the relative change in the length ($\delta L/L$).

The `Gage Factor' is a constant for each gage if it stays in the normal temperature and distortion range.

The most used metal for the strain gages are the following:

- Nickelchrome; with a factor of about 2.25
- Platinum; with a factor of about 3.7
- constantan; with a factor of about 22

Nevertheless, it is impossible to use directly the resistance variation of the strain gages, which is too small. The strain gages are placed in a Wheatstone bridge to amplify the signal.

There are many kinds of bridges (quarterbridge, halfbridge, fullbridge), which differ according to the number of active parts in the bridge.

In our case we use a halfbridge, which offers better temperature compensation than the quaterbridge.

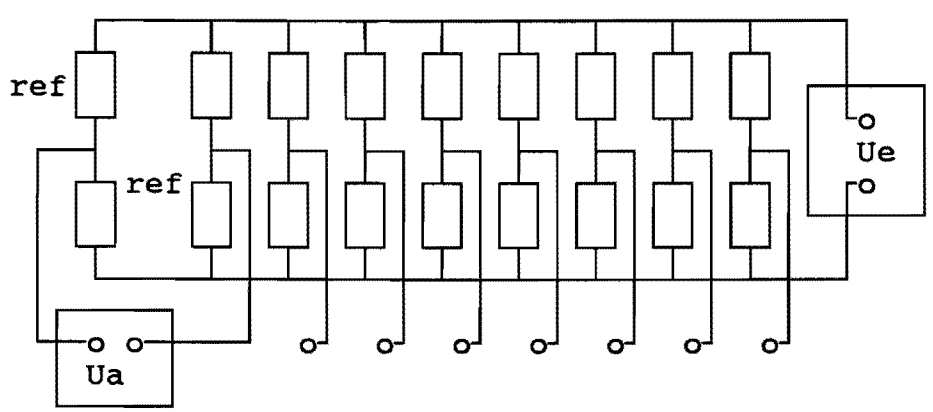


fig.11: Schematic of the bridges

III.6.1.3 Electronic Description

Some electronics have been included inside the torque sensor, the aim of this electronic card is to amplify the signal issued from the Wheatstone bridge.

In this goal an instrumental amplifier (INA 101 from Burr-Brown detailed in the appendix) is used, it is exactly designed for such an application.

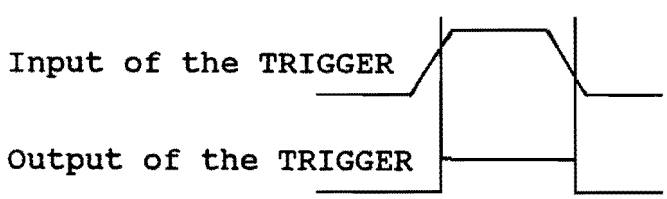
With Ue= 5v, we can expect an Ua,max= 8 mv; so the amplification has to be at least of one thousand times to supply an optimal signal for the interface.

It seems that this electronic card is not totally perfect, because a lot of noise is present on the outputs. This noise comes mainly from the interaction between the printed circuit board and the structure of the sensor.

Much progress could be achieved in this field to increase the accuracy and the reliability of the acquisition chain.

III.6.2 The Hall Switches

In order to determinate the limits of the movement, either the rotation and the translation are equipped with two hall effect switches. Two Nand Schmitt Trigger integrated circuits have been used to clean up the wave form.



III.6.3 The Position Sensor

The same for the linear and the circular part of the robot, the incremental encoder (LIDA 360 from HEIDENHAIN) uses a scale tape which can either be placed on a circular or a planar area.

This is an incremental transducer and allows the sense discrimination. The precision of this sensor may be increased by 25 with an electronic interpolation and impulsformer (EXE 702 from HEIDENHAIN).

III.6.4 The Power servo Amplifier

Axodyn power servo amplifiers Series 05 LV comprise continuously operating transistored amplifiers. The amplifier is a multi-stage design for linearity and gain reasons. Output current and output voltage are for optimal controlled through the power stage in a parallel process.

The power section of the 05 LV comprises driver and power stages. Each transistor is provided with its own emitter resistance to improve current distribution and the symmetrical arrangement allows the unlimited four quadrant operations

This device includes also a speed regulator which is a tacho control, but it isn't used in our application.

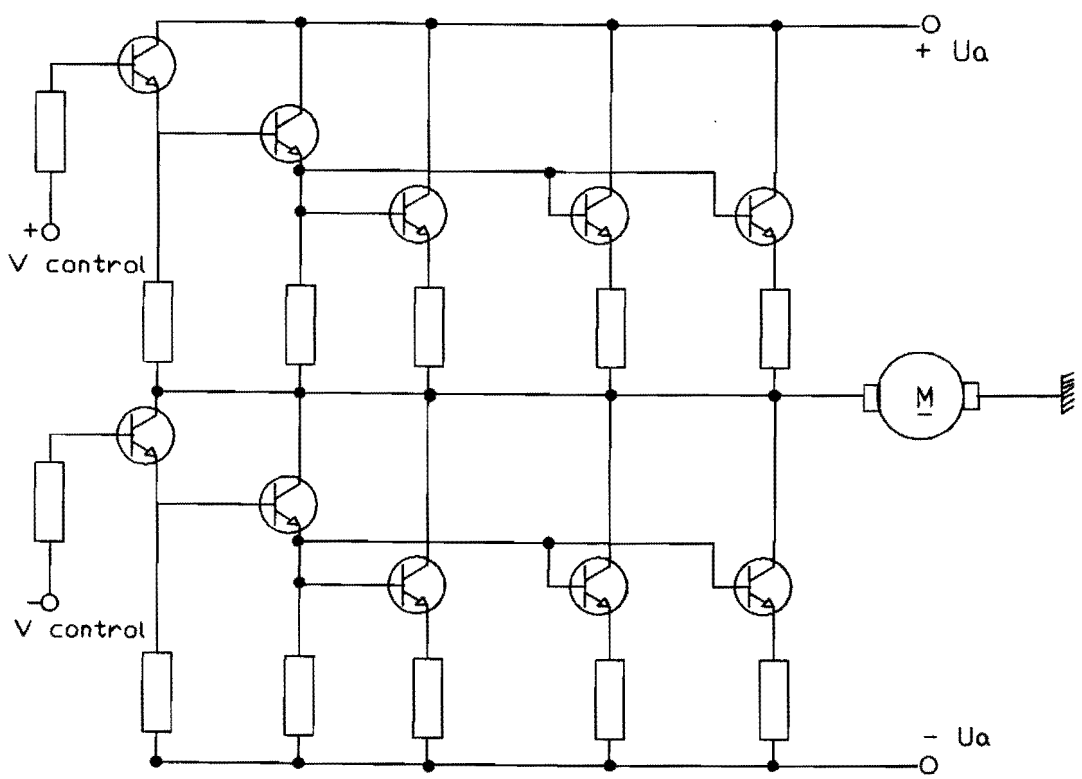


fig.12: Power Section of the Axodyn Power Servo Amplifier

IV THE MULTIPROCESSOR SYSTEM

We have to keep in mind that the final goal of the control is to operate a robot in real time. In this way a multiprocessor system or more precisely a parallel processor has been developed, it includes four single board computers, all from Intel, in order to maintain a compatibility between the boards and to benefit from the long experience of the supplier.

In the following paragraphs, we will see the various boards used and the way they communicate together and with the hardware environment. It is obviously not possible to give a complete description of the system, but for more information the references for the individual reports are in the bibliography.

IV.1 The communication

This is one of the most crucial parts of the system, since we use many boards to share the operation of the control, as well as the fact that the effective command to the robot includes knowing the actual and the future state of this one.

As stated in the introduction, we use four single board computers; one for the control of the translation, one for the control of the rotation, another one for the acquisition of the data from the interface and finally the last one called the master, which used to control the other boards and has no communication with the robot.

But we also use a multibus system to establish the data communication between the boards and a RAM board to store the data.

It is now more obvious that with so many subsystems, we have many possibilities to organize the communication and the computer architecture.

The schematic on the next page explain the system implementation, but before going further the references of the various boards must be given:

- Master computer: iSBC 186/03
- Translation computer: iSBC 186/03
- Rotation computer: iSBC 186/03
- Sensor computer: iSBC 86/12
- Ram board: iSBC 028A

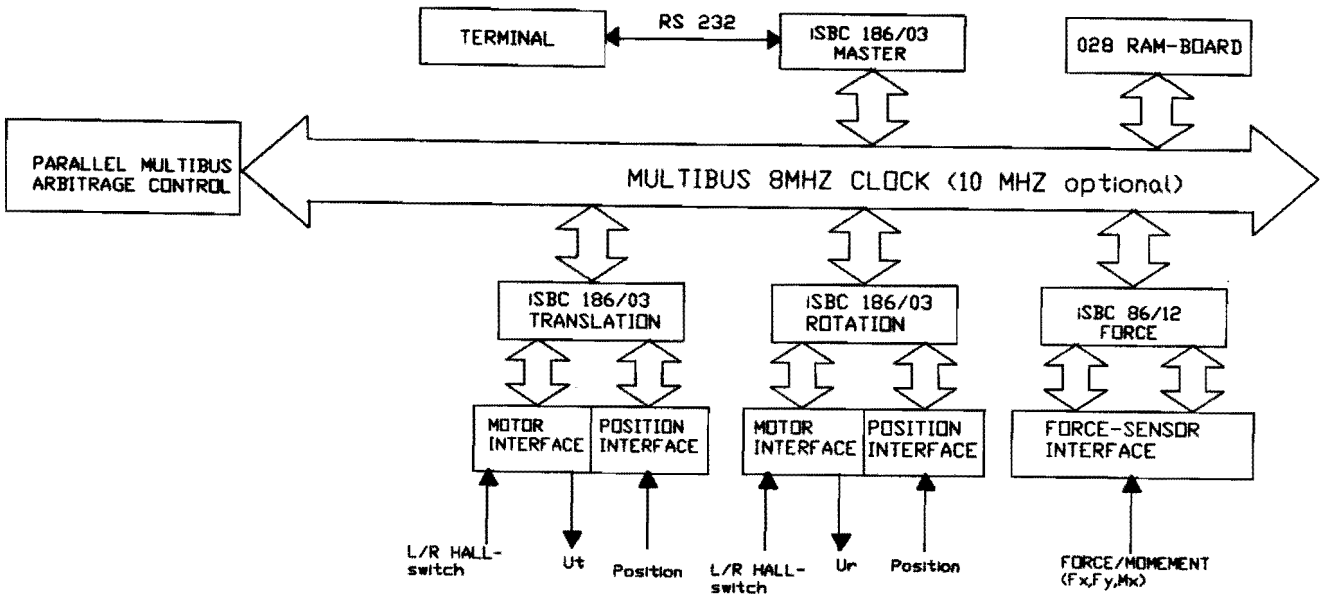


fig.13: System Architecture

IV.2 Multibus System

IV.2.1 Definition

The multibus system requires a great deal of effort to understand completely. The multibus system is a standard of communication introduced by Intel in 1986. It is also known under the name of IEEE 796.

This bus architecture is the conceptual foundation and physical framework for interfacing the various components of the Intel family. This family includes single board computers, memory expansion boards, a broad array of I/O expansion products, packaging products, microsystems software and integrated microsystems.

It contains all the necessary signals to allow the system components to interact with one another. This system is based on the Master-slave concept. The `Handshaking' protocol between masters and slaves allows computer of different speeds to use the multibus interface and also allows data rates of up to five million transfers per second. The multibus system bus can support multiple master device on a back plane and can directly address up to 16 megabytes of memory.

Its interface structure consists of 24 address lines, 16 data lines, 12 control lines, 9 interrupt lines and 6 bus exchange lines.

To sum up, the multibus interface is included on each single board and could be seen by the user as a parallel interface, needed only proper configuration before use.

IV.2.1 The Multibus arbitration

For the multibus arbitration, we use a parallel priority resolving technique which uses a separate bus request line (BREQ) for each arbiter on the multi master system bus. Each BREQ line enters into a priority encoder which generates the binary address of the highest priority BREQ line which is active.

The arbiter receiving priority (BPRN true) then allows its associated bus master onto the multi-master system bus as soon as it becomes available. If a higher bus master is requesting the bus it has to wait for the present instruction to be finished.

However a lower bus master can keep the multibus by using the lock instruction which disables the other masters to use the bus.

There are two other ways of arbitration:

- The serial arbitration: The BPRO (Bus Priority Out) of each master bus is connected to the BPRN line of the next priority master bus.
- The rotational priority resolution: This technique is similar to the parallel resolving technique except that the priority rotates. This allows each arbiter an equal chance to use the multi-master system bus.

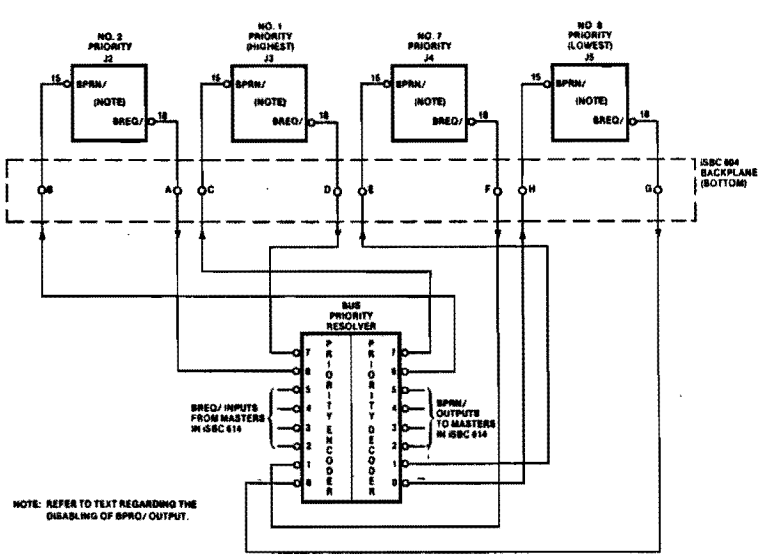


fig.14: Parallel Arbitration

IV.3 The RAM Memory Board: ISBC 028A

The iSBC 028A random access memory board provides a dynamic memory storage capacity of 128 K bytes. This RAM memory interfaces directly with the bus master via the multibus interface in any 8 or 16-bit ISBC operating board.

- Access time: read --> 500 ns max
  write --> 343 ns max
- Cycles times : Read, Write and Refresh in 608 ns max

This RAM board is used to store the common data; forces, moments position registered, position calculated ...

IV.4 Single Board Computers

IV.4.1 iSBC 186/03

The iSBC 186/03 is a general purpose, 16-bit computer system on a multibus-compatible printed circuit board. The board supports the high-speed memory execution bus (iLBX bus) for local memory expansion. The board is designed to interface to the small computer interface (SCSI) with some reconfiguration and some optional components.

The specifications of this single board computer are the followings:

- iAPX 186 (80186) high integration microprocessor with a speed of 8 MHZ.
- iAPX 86/30 (80130) operating system processor.
- Eight byte-wide memory sites for EPROM, EEPROM, SRAM, iRAM, or NVRAM.
  Configurable memory capacity: up to 32K bytes of RAM and up to 256K bytes of ROM memory on the board.
- iSBC 341 memory expansion multimodule board compatible, expanding RAM to 64K bytes.
- iLBX local memory expansion for high-speed transfers to/from iLBX memory expansion board.

- 27 interrupt sources on-board using the 80186, 80130 and 8259a interrupt controllers and the 8274 serial controller.
- Two serial I/O channels controlled by an 8274 multi-protocol serial controller (MPSC).
- General purpose parallel interface; can be configured to the SCSI or Centronics interface by adding PALs.
- Two iSBX bus interface connectors.
- Master capability.
- Dedicated front panel interface connector.

After this short introduction, the user could find more informations in the iSBC 186/03 Single Board Computer Hardware Reference Manual (Order Number: 146414-001).

IV.4.2 iSBC 86/12

We will examine this single board computer more in detail, because it is directly used for the acquisition of the interface's data.

It would be difficult to explain the choice of the 12 board for this particular use, because it was the responsibility of previous students.

The major quality of this board is the Dual Port RAM, this RAM is either addressed by the CPU and another bus master controlling the multibus. So the communication between the CPU and the other single board computers can be done without stopping the CPU. Any device can put the data in the Dual Port RAM which can be accessed at any time. In this particular use, we call the iSBC 86/12 a slave RAM.

The iSBC 86/12 single board computer is controlled by an Intel 8086 16-bit Microprocessor (CPU). The CPU includes four 16-bit registers that may also be addressed as eight 8-bit registers.

The CPU contains also two 16-bit pointer registers,two 16-bit index registers, four 16-bit segment registers.

The CPU instruction set supports a wide range of addressing modes and data transfer operations, signed and unsigned 8-bit and 16-bit arithmetic including hardware multiply and divide, and logical and string operation.

Four IC sockets are included to accommodate up to 16K bytes of ROM.

The iSBC 86/12 provides 24 programmable parallel I/O lines implemented by means of an Intel 8255A PPI (programmable peripheral interface) chip.

The RS232C compatible serial I/O port is controlled by an 8251A USART (universal synchronous/asynchronous receiver/transmiter) chip from Intel.

Three independent, fully programmable 16-bit interval timer/event counters are provided by an Intel 8253 PIT (programmable interval timer) chip.

Concerning the interrupts the CPU has non-maskable interrupt (NMI) and maskable interrupt (INTR). The NMI interrupt is intended to be used for catastrophic events such as a power failure that requires immediate action of the CPU. The INTR interrupt is driven by an Intel 8259A PIC (programmable interrupt controller) chip which, on demand provides an 8-bit identifier of the interrupting source.This 8-bit identifier is multiplied by four to provide a pointer for the interrupt service routine.

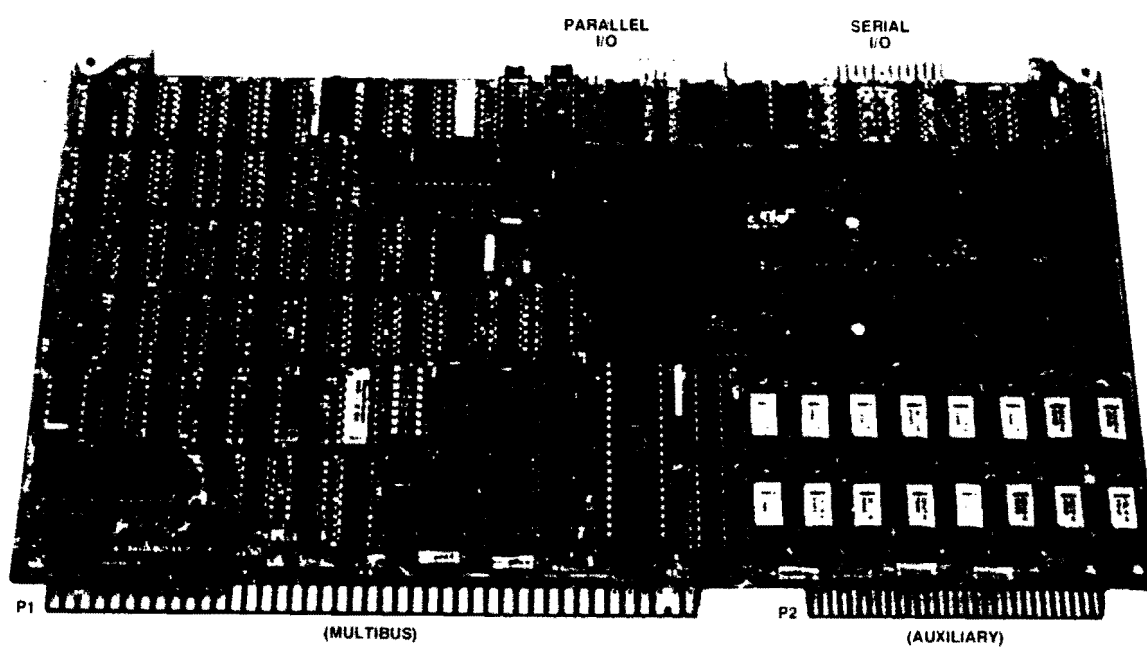


fig.15 intel Single Board Computer 86/12

IV.5 Memory Addressing Scheme

It is important to understand the memory segmentation for the use of a 16-bit microprocessor.

Memory adresses are logically subdivided into segments of 64K bytes each, which can be allocated to code, data or stack. Each segment must start at an address which is evenly divisible by sixteen. At any time the contents of four of these segments are immediately addressable (current code segment, current data segment, current stack segment, current extra segment). Since the effective address of a 16-bit microprocessor is being 20-bit and the segment used is only 16-bit; we must use an other 16-bit word called offset to calculate the address.

Thus, a 20-bit address is constructed by adding the 16-bit offset address to the 16-bit segment address with four lower zero bits appended. That is, they are left shifted four places which effectively multiplies the segment value by sixteen.

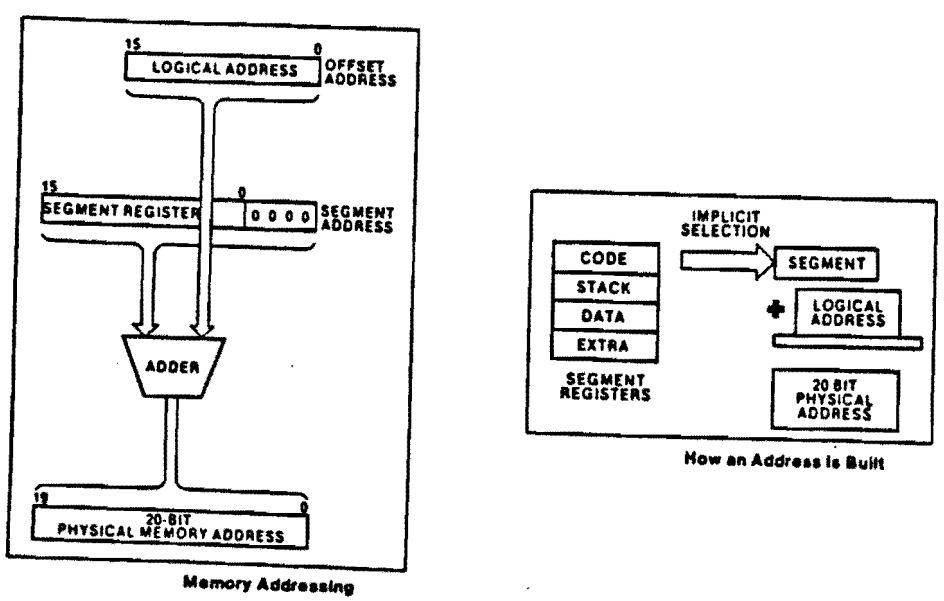


fig.16: Memory Segmentation

V THE SOFTWARE

This is a very important part for proper operation of the interface. We will see the software development, the structure of the program and explore different parts of the software.

It is unnecessary to explain in detail the program. The user is left to examine the program listing and documentation.

V.1 Program Development

At the beginning, we didn't know if the software should be written in assembly or in pascal, as both languages have advantages and drawbacks. It appeared more useful, quicker and easy for the debugging to write a first version in pascal.

In any event, it will be easier to implement an assembly program by using the algorithm of the pascal one.

Learning to use the Pascal 86 took little time, and the time was more often spent on configuration problems (PPI,PIC,PIT) than on the specifications of this language, which was quite close to the standard pascal.

We intended to do a program as modular as possible, in order to allow further development.

Many problems came from the use of the monitor IAPX 86-88 which wasn't adapted to the interrupt of the intel Single Board Computer 86/12. The use of the iSDM 86 Monitor V 1.0 solved all these problems.

The last logical step was the simulation of the interface which showed the limits of the software. The maximum speed obtained was about 10 KHZ.

V.2 Program Structure

As explained in the previous chapter, this program has been built in four modules.

The program itself called MAIN; calls the routines of other modules, set the interrupts and runs the acquisition. This module contains also the interrupt routines which was initially put in a separate module.

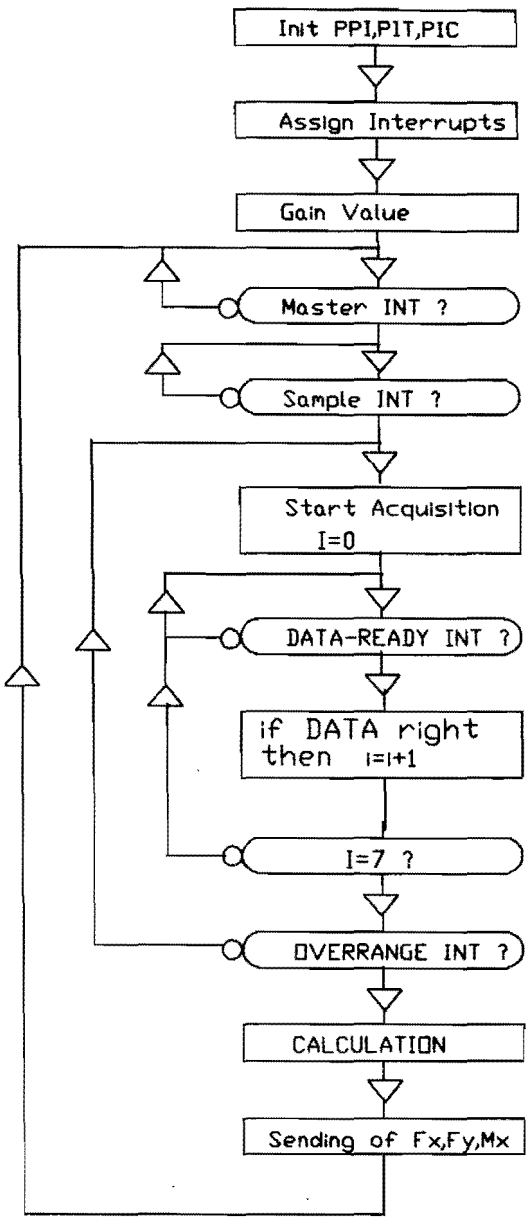
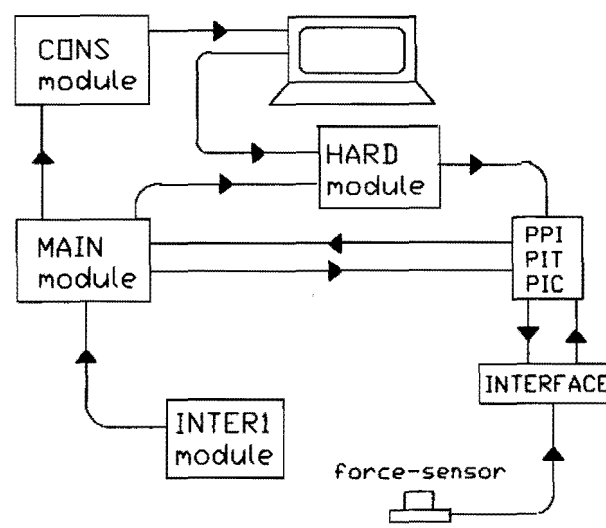


fig.17: FLOWCHART & MODULES COMMUNICATION



The module used for all the public declarations needed in the multi-module structure is called INTER1.

The hardware initialisation modules (PPI,PIC,PIT) is named HARD.

To facilitate the communication between the operator and the computer we introduced a module under the name of CONS.

The idea of using an interface module (INTER1) is useful for the development of the program. A change in data only requires the change of the interface specifications.

In the future, the communication between the computer and the operator should be run by the master, meaning that the communication module would be suppressed and the data should be read at a specific address in the RAM board.

V.3 Calculations

We have many way to do the calculation, but it is obvious that we have to consider that the acquisition is not continuous. Many algorithms permit such calculations, but as a first step we use an easy one; a linear interpolation. The precision could be increased by using a most powerful method (higher level of integration), but this better precision would be to the detriment of the system speed.

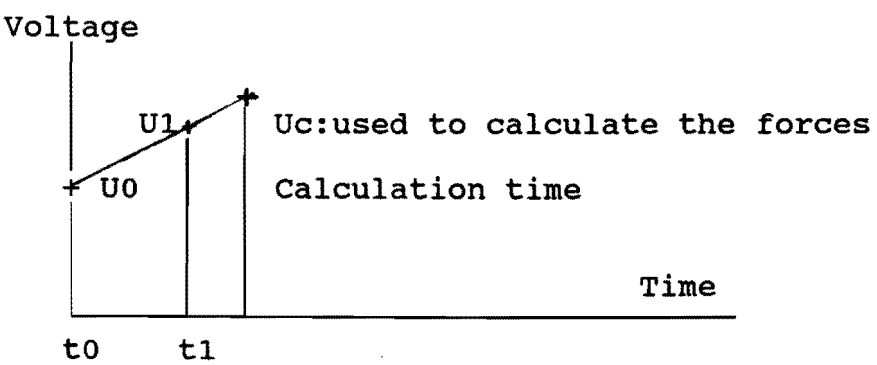


fig.18: technique of calculation

However, it is also important to consider that the eight voltages are measured at different moment. So, the chances to get results close to the reality decrease if we consider the first channels selected. To equilibrate the approximation of the various voltages, the order of acquisition should change after the end of the calculation of the constraints. in our case, we can not change the order because of the counter.

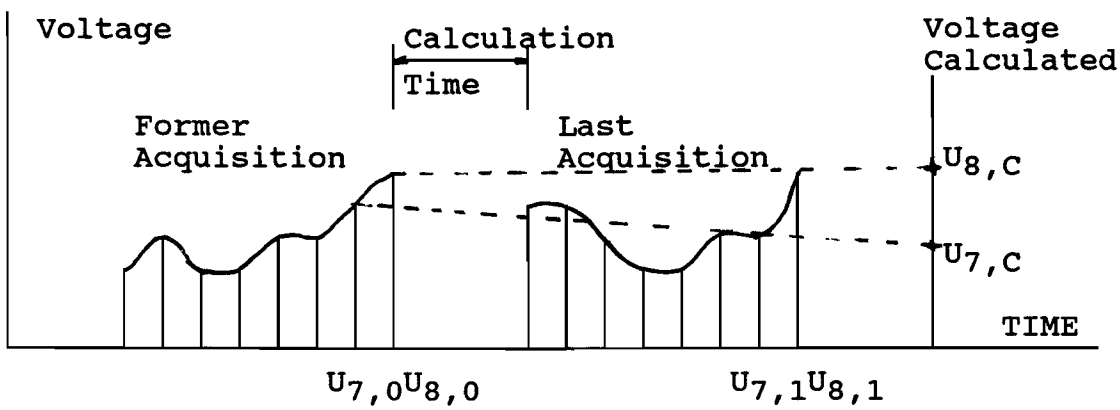


fig.19: differences between the various approximations

V.4 interrupt procedures

In our program, only two interrupts are generated by the interface:

- DATA_READY: connected to the busy line of the SDM 856 (see chapter interface) informs the computer that the conversion is finished.
- OVERRANGE: connected to the overrange detector of the interface, it is set if one of the analog signals is out of the range [-10v,+10v].

The other interrupts are generated by the master computer (see next chapter for the definition of the modes):

- TEACH_INT: means that the teach mode is activated.
- TEST_INT: Occurs when some tests of the interface, the RAM board or the various computers are done.
- REPLAY_INT: starts the replay mode.
- SAMPLE_INT: generates the sample time for the acquisition.

V.5 Definition of the Various Modes

V.5.1 Test Mode

In this mode, all the parts of the control system may be checked; the RAM board, the interface, the counters and the DACs (counter and DAC from the rotation and translation interfaces), this mode is not always used but might prevent many troubles if a improper operation of the robot is detected.

V.5.2 Replay mode

When activated, this mode makes the robot copy the trajectory registered in the teach mode. The robot is able to do the same movement even if a greater force is applied on it; which is the objective of the method. We teach the trajectory without constraints and then the robot should be able to move in the same way, whatever its load.

V.5.3 Teach mode

In this mode, the teaching method already defined is used, the force sensor is obviously used to record the path.

V.6 Peripherals

V.6.1 PPI

The PPI includes three ports (port A, port B and port C) which offer many possible configurations. It has 24 I/O lines which may be individually grouped in 2 groups of 12 and used in 3 major modes of operation. In the first mode (mode 0), the lines are programmed by group of four Input or Output lines. In mode 1, the second mode each group is programmed to have eight Input or Output lines, the four lines remaining are used for the handshaking and the interrupt control signals.
Only the port A is affected by the last mode (mode 2), it allows eight lines to be configured either as Input and Output, that is to say bidirectional lines. Five other lines are used as control lines borrowing one from the other group.To programm the PPI, only the writing of a control word into the 8255A is required.

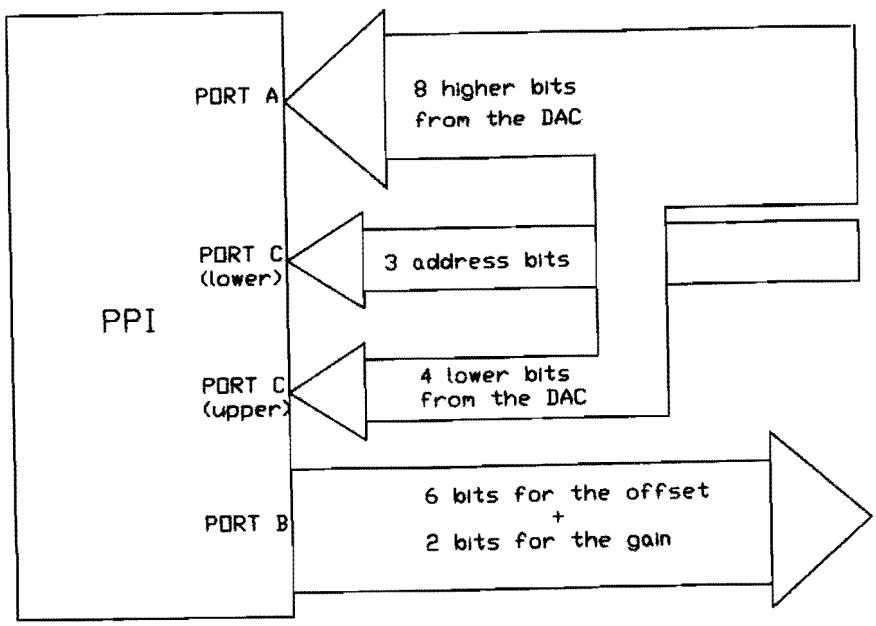


fig.20: Configuration of the PPI

V.6.2 PIT

A timer is a very useful component to generate a sample rate or a simple clock without involving the CPU.

The Intel 8253 is organized as three independent 16-bit counters, each with a count rate of up to 2 MHZ.

The counters may be either programmed to count in BCD or in binary.

Each counter can be connected to one of the three clock rates available: 153.6 KHZ, 1.23 MHZ or 2.46 MHZ. We have also the possibility to connect the output of one counter to the clock of another one.

A gate allows an external device to trigger the counter if this one is so programmed.

Five modes of operation for each counter are programmable by the software with simple I/O operations. The three counters may be programmed individually at any time, It requires a control word and a count value for initialisation. Usually consisting of two bits, the count value may be reduced to only one; MSB or LSB.

Mode 0: Interrupt on Terminal Count. The output will be initially low after the loading of the control word and stay low until the terminal count is reached. This occurs assuming that the gate is connected at a `1' Level.

Mode 1: Programmable One-Shot. The output will go low on the count following the rising edge of the gate input and remain low until the terminal count.

Mode 2: Rate Generator. The output is initially high, then it goes low for one cycle. The process then repeats.

Mode 3: Square Wave Generator. high during on half of the count and low during the other half. for odd count the output stay high for (N+1)/2 of the count and low for (N-1)/2 of the count.

Mode 4: Software Triggered Strobe. The output will go low for one clock period after the terminal count which is initialized by the loading of the counter.

Mode 5: Hardware Triggered Strobe. The counter will start counting after the rising edge of the gate input and the output will stay low for one clock period at the end of the count. The counter is retriggerable.

V.6.3 PIC

The Intel 8259 handles up to 8 vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored interrupts.
The 8259 is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirement.
To use this device two types of command words should be defined; the Initialization Command Words (ICWs) and the Operation Command Words (OCWs).

ICW1 indicates if that PIC takes place in a multi-pic configuration or not and defines the level or the edge triggered input for the interrupts.
ICW2 gives the address of the first vector interrupt.
ICW3 is only used for a slave PIC and indicates to which master interrupt it is connected.
ICW4 defines whether or not an EOI (End Of Interrupt) is generated and if the Fully Nested Mode is used (Interrupt requests are ordered from 0 trough 7).

Concerning the Operation Command Words, only OCW1 has to be clearly defined because it sets the interrupt mask. The others can be forgotten in most of applications.

V.7 The development system

V.7.1 Presentation

The Intellec serie III micro-computer development system is a useful tool for designing microcomputer software for the IAPX 86-88 processors. We can write programs, debug them, link them, locate them and run them on single board computers or on the system itself. We can connect an emulator for running our programs in the hardware environment.

This system offers the possibility to write source programs in high level languages: Pascal, Fortran, PLM and Assembly. These languages are special Intel's versions (PASCAL 86, FORTRAN 86, PLM 86, ASSEMBLY 86) which are quite close to the standard versions.

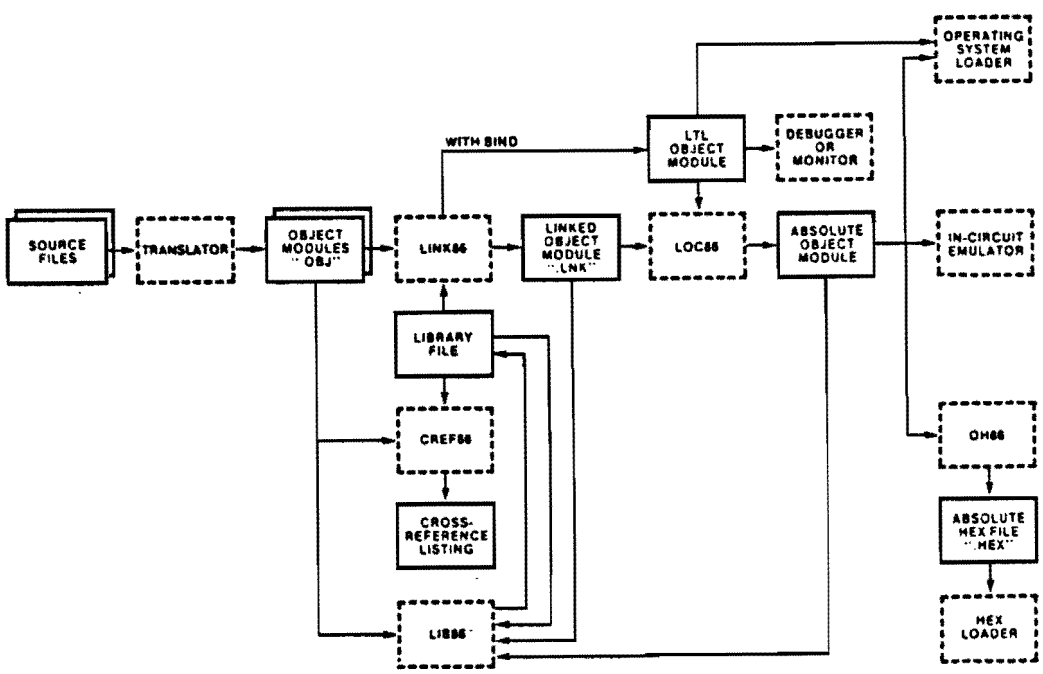


fig.21: Drawing of the System Development

V.7.2 Compiling and Linking Operations

The Pascal 86 compiler is very powerful and efficient, it generates a file with the same name than the module compiled, but with the LST extension.

The interface module can not be compiled alone, nevertheless it is compiled with all the other modules.

The Linker is also a powerful tool to link all the Object files, as well Pascal or Assembly object files. However, the linking can not be done without using the libraries required.

The files P86RN0.LIB and P86RN1.LIB are both always required, when using real number, the file 87NULL.LIB is necessary and the use of several modules involves the file RTNULL.LIB.

After the linking task, the object file has to be located; the DATA segment, the STACK segment, the CODE segment and the starting address must be defined.

Then, the located file can be converted in an HEX file to be put in an EPROM or to be downloaded from the development system.

VI. CONCLUSION

The management of a project abroad is totally different from a project done in our own country, because in addition to the technical work there is the every day life which requires an adaptation and some changes in our habits.

Most of Dutch people are fluent in English and German, so the communication is always possible.

The technical work was interesting, not very new for me, but anyway it was a good experience.

One of the major problem was the fact that many students have worked on the robot, and some of them didn't reach satisfying results, so their work had no conclusion. I wanted to avoid such a situation, that's why I wrote a report as complete as possible to allow any further development. The conclusion on my work are not easy to reach, because the interface wasn't tested in real conditions.

However, I'm very optimistic regarding the performance of it.

The atmosphere of working was pretty good, and the other dutch students helped me a lot for my integration among them.

I improved my English a lot and I discovered a country which is really oriented to the foreign countries and well prepared to the future European Common Market.

APPENDIX CONTENTS

Appendix I: French Summary
Appendix II: Practical Informations
Appendix III: Schematic of the Interface
Appendix IV: SDM 856
Appendix V: UAF 21
Appendix VI: AD 526
Appendix VII: INA 101
Appendix VIII: DAC 80
Appendix IX: SN 74100, SN 74193
Appendix X: Motor, Power Amplifier and Encoder
Appendix XI: Listing of Main.src
Appendix XII: Listing of Inter1.src
Appendix XIII: Listing of Hard.src
Appendix XIV: Listing of Cons.src
Appendix XV: Listing of Location File.
Appendix XVI: References

Appendix I

RESUME EN FRANCAIS

Durant ces cinq mois, j'ai apporté ma contribution a l'élaboration du control d'un robot qui comporte deux degrés de liberté (Rotation et Translation). La finalité de ce robot est de répéter des trajectoires enregistrées préalablement. L'originalité de ce robot consiste dans l'utilisation d'un capteur de force tridimensionnel qui est précisement utilisé pour l'enregistrement des trajectoires.

Mon champ d'activité s'est surtout situé au niveau de l'acquisition des efforts appliqués sur le robot et mesurés par le capteur de force. A cet effet, j'ai conçu une interface ainsi que le programme qui la gérait.

Mon travail s'est déroulé en plusieurs étapes; au départ il m'a fallut acquerir certaines connaissances sur le robot lui même, ensuite je me suis occupé de la conception de l'interface et finalement de la mise au point du programme de gestion de l'interface. Le programme a pu être testé dans de bonnes conditions, mais l'interface ayant été achevées trop tard, je n'ai pas pu effectué tous les tests requis. Mon travail sera donc conclu par un autre étudiant; j'ai donc apporté un soin tout particulier a la rédaction de mon rapport.

Ce travail n'était pas d'un niveau technologique et scientifique très élevé, mais la nécéssité de communiquer en Anglais, de lire des articles en Anglais et de dépendre d'autres services apportent des difficultés qui s'estompent peu à peu, mais restent malgré tout un obstacle à la conduite de projet plus conséquent dans un temps limité.

Sur le plan personnel, ce stage a été une experience très enrichissante; découverte d'une autre culture, amélioration de mon Anglais, mises en pratique de connaissances théoriques et surtout éveil à un esprit international.

Appendix II

PRACTICAL INFORMATION

1. Use of SUBMIT program

This program allows the user to create a file called batch file which may prevent to compile,link and locate all the files separately. This BATCH file is called BATCH.CSD in our example (see following page). By this way, if some modifications are brought to any modules, the user don't have to wait that one task is finished to start the following one; he has just to run the SUBMIT program with the syntax below:

SUBMIT :Fx:BATCH.Extension(Name1,Name2, Name3,...)

Fx is the drive where the BATCH file is put

Name1,Name2,.. are the source files.

In our example, we only use three source files, but the batch file can run with five source files. As we can see on the listing some lines start with ';' separator; this means that the line won't be interpreted by the SUBMIT program.

2. How to start the program ?

With the iSDM86 Monitor, a located file is required to be downloaded from the system development to the Single Board. For this task a program called iSDM86 is supplied by Intel, but the various operations have to be done in a certain way.

The transfer of the located file requires the same baud rate between the system development and the Single Board, so when the iSDM86 file is activated, then the Single Board has to be reseted. When the Single Board is ready, it print on the screen:

```
iSDM 86 Monitor, V1.0
Copyright 1983 Intel Corporation
.
```

Then the file can be loaded in the on board RAM with the instruction: L:Fx:Name.STA

x being the drive where is the file

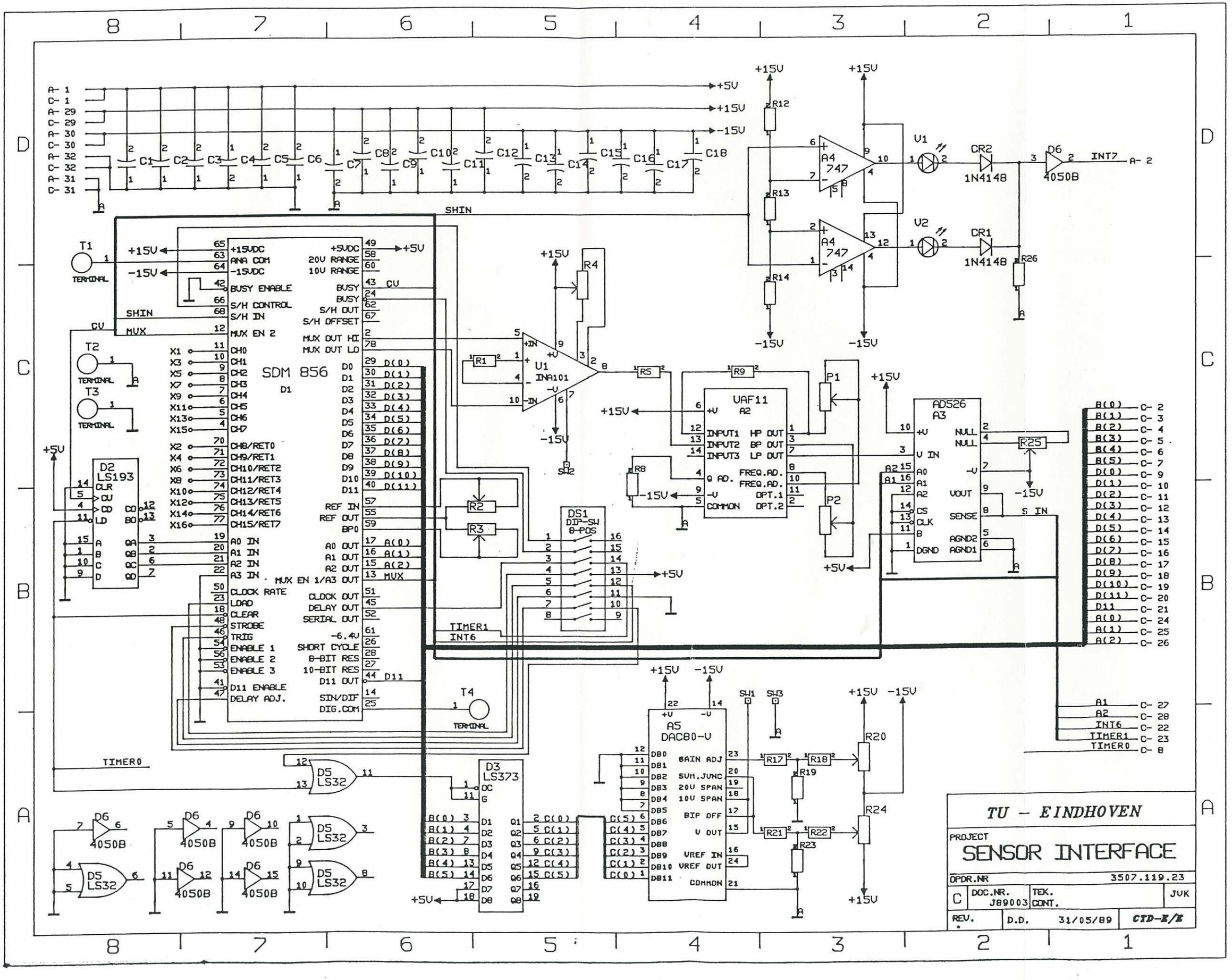
Name is the name of the file

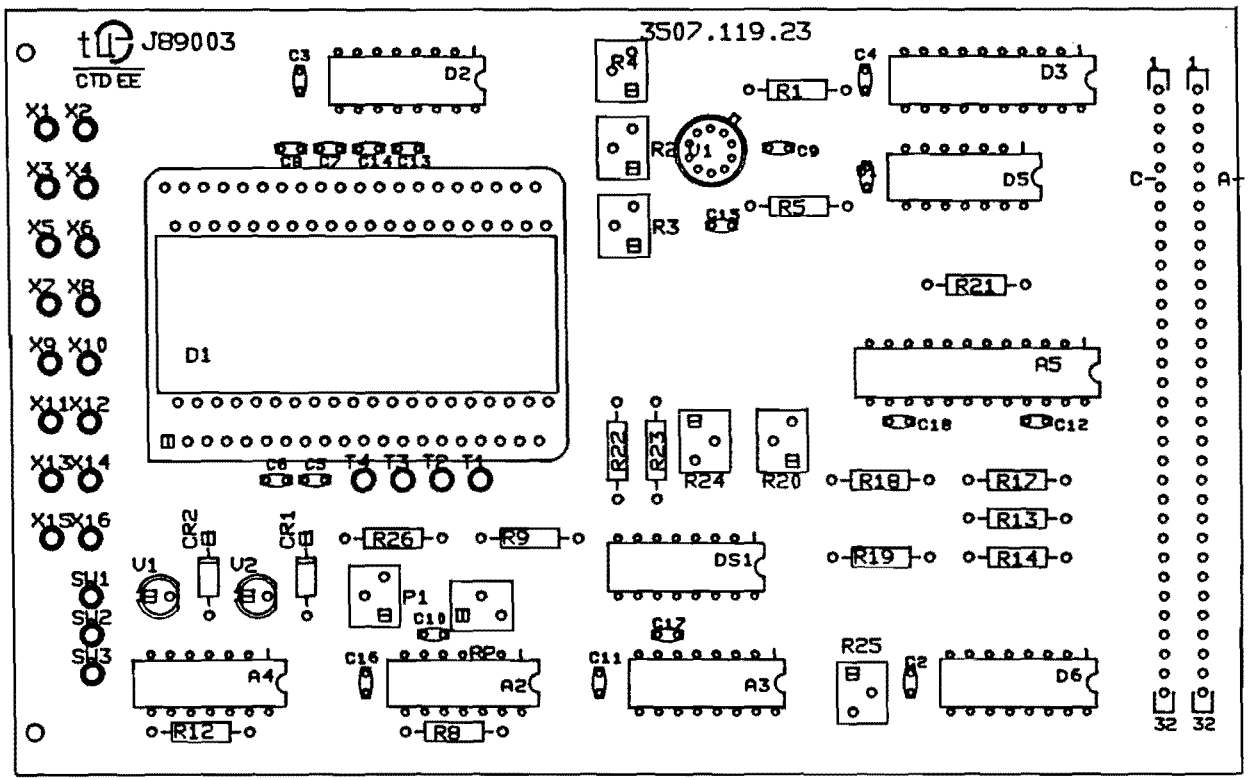
STA is the extension given the batch file (see previous chapter). When the monitor is ready, it prints '.' on the screen.

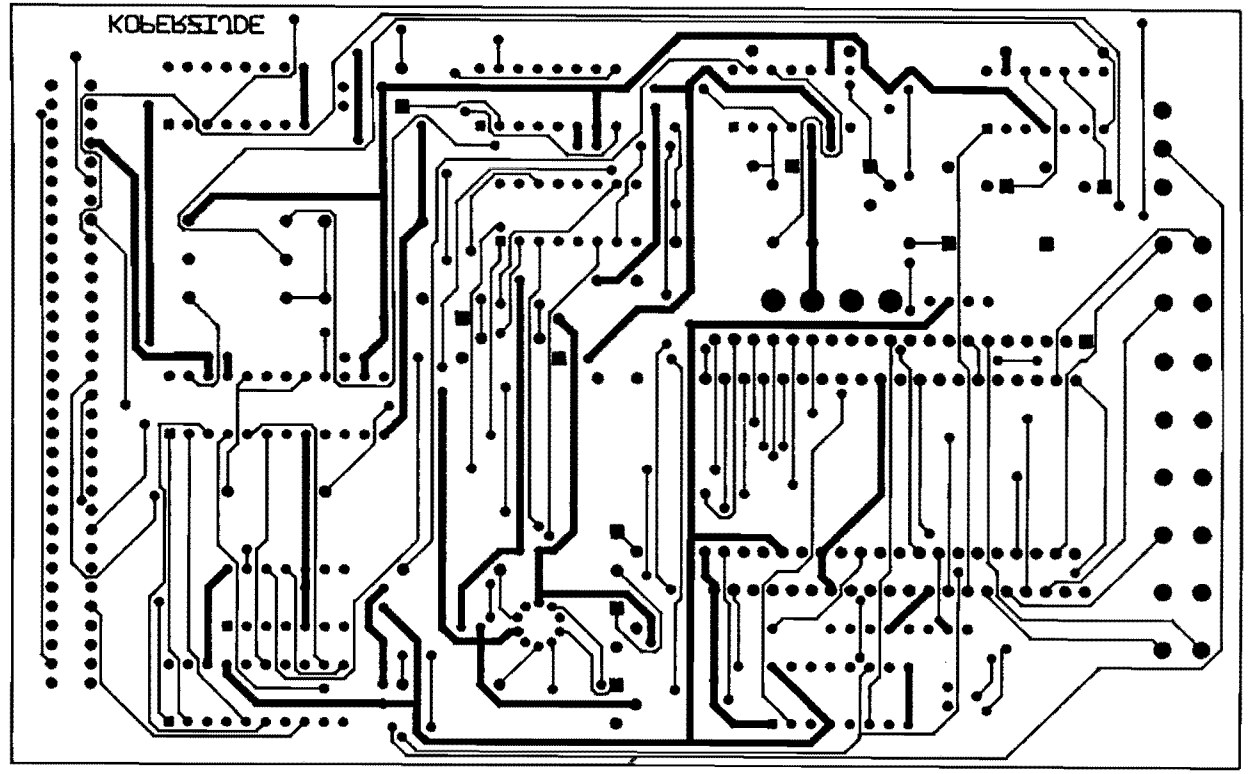
The program is able to run till this moment, the instruction 'G' performs this task. To stop the program, the 'CNTL C' Key can be used. Then the 'G' instruction starts the program again or the 'E' instruction permits to the user to Exit the current program and return to the ISIS-II environment.

```
:file:BATCH.CSD
;
;!-------------------------------------------------------------------!
;!                                                                   !
;!       COMPILER & LINKER & LOCATER (& HEXCONVERTER)                !
;!                                                                   !
;!                                                                   !
;!                                                                   !
;!       Put     in drive 0 de systeemdisk,                          !
;!               in drive 1 de pascal86floppy,                       !
;!               in drive 4 de program.                              !
;!-------------------------------------------------------------------!
;
;*******************************************************************
;*          Compiling of the source files                          *
;*******************************************************************
;
RUN
:F2:PASC86 :F1:%0.SRC
:F2:PASC86 :F1:%1.SRC
:F1:PASC86 :F4:%2.SRC
;:F1:PASC86 :F4:%3.SRC
;:F1:PASC86 :F4:%4.SRC
;
;*******************************************************************
;*           linking of the object files                           *
;*******************************************************************
;
LINK86 &
:F4:%0.OBJ, :F4:%1.OBJ, :F1:%2.OBJ, :F4:%2.OBJ, &
;:F4:%3.obj, :F4:%4.obj, &
:F1:P86RN0.LIB, :F1:P86RN1.LIB, &
:F1:RTNULL.LIB, :F1:87NULL.LIB, :F1:cel87.lib, &
:F1:SDMIOL.LIB, :F1:MBUS86.LIB, :F1:SBCIOS.LIB TO :F4:%1.LNK
;
;*******************************************************************
;*          locating of the link file                              *
;*******************************************************************
;
LOC86 &
:F4:%1.LNK TO :F4:%1.STA &
INITCODE (1000H) &
ORDER (CLASSES(CODE,DATA,STACK)) &
ADDRESSES (CLASSES (CODE(1400H), DATA(2400H), STACK(3400H)))
;
;*******************************************************************
;* INITCODE,ORDER and ADDRESSES have to be compatible              *
;*******************************************************************
;
;*******************************************************************
;*         CONVERSION to HEXA-DECIMAAL formaat                     *
;*        creation of a file with .HEX extension                   *
;*******************************************************************
;
;OH86 &
;:F4:%1.STA TO :F4:%1.HEX

EXIT
```

Appendix III

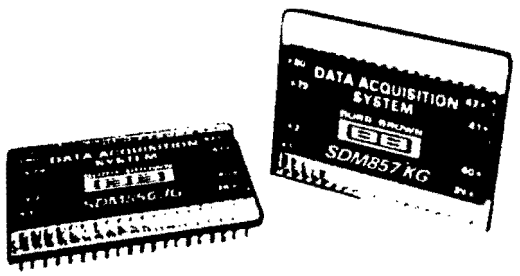



J89003
CTD EE
3507.119.23
D1
D2
D3
D5
A5
A4
A2
A3
D6
DS1
P1
V1
V2
SH1
SH2
SH3
R1
R2
R3
R4
R5
R9
R12
R17
R18
R19
R20
R21
R22
R23
R24
R25
R26
C3
C4
C9
C10
C11
C12
C13
C14
C15
C16
C17
C18
X1
X2
X3
X4
X5
X6
X7
X8
X9
X10
X11
X12
X13
X14
X15
X16
C
A
32
32


KOBERSIDE

Appendix IV

BURR-BROWN®



SDM856
SDM857

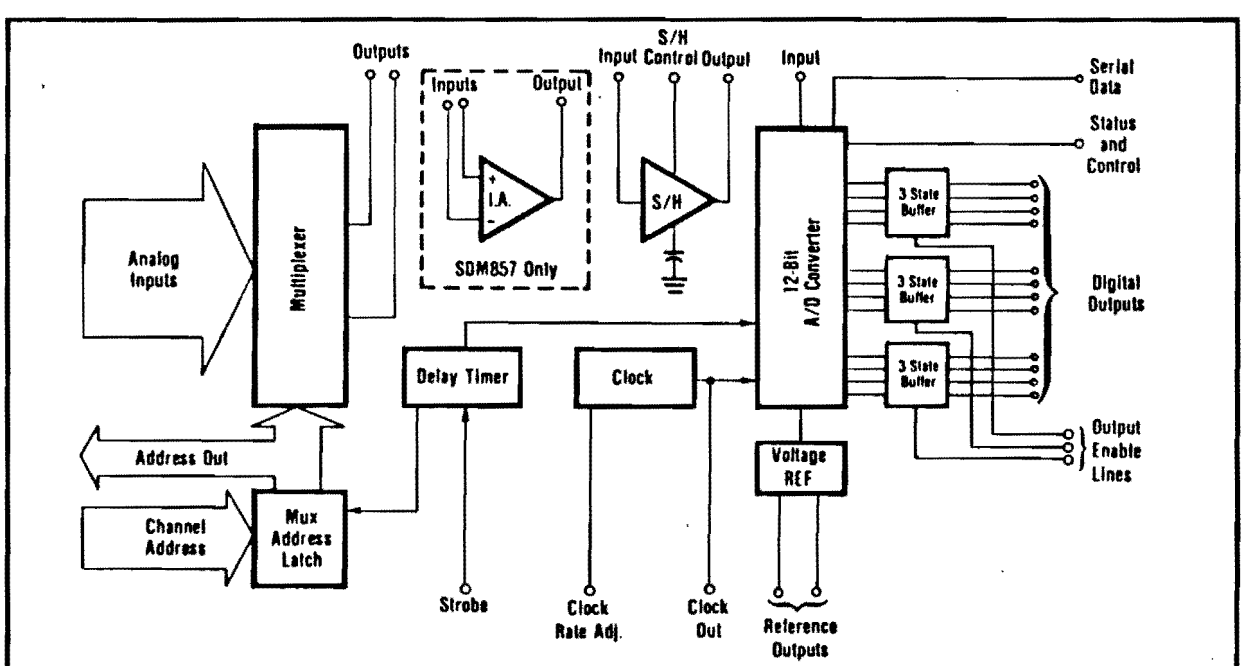
HYBRID DATA ACQUISITION SYSTEM

FEATURES

- MINIATURE SIZE
- LOW COST
- 12-BIT, ±0.012% LINEARITY ERROR
- INSTRUMENT AMP OPTION
- LOW LEVEL INPUTS (SDM857)
- SELECTABLE 16 SINGLE, 8 DIFFERENTIAL INPUTS
- THREE-STATE OUTPUT BUFFERS
- THROUGHPUT RATES (SDM857 Overlap Mode)
  - 8-Bit Accuracy: 70kHz
  - 10-Bit Accuracy: 32kHz
  - 12-Bit Accuracy: 29kHz

DESCRIPTION

The SDM856 and SDM857 are complete data acquisition systems contained in a miniature 2.2" x 1.7" x 0.22" ceramic package. These systems offer all the functions available in large modular data acquisition systems and are available with an optional internal instrumentation amplifier (SDM857). Inputs as low as ±10mV can be accepted by the SDM857; thermocouples, strain gages, and other low level signal sensors don't require external signal conditioning. Both models are fully expandable from the basic 16 channel single-ended or 8 channel differential input capability. Digital resolution is 12 bits with accuracy of ±0.024% at a throughput rate of 29kHz (SDM856KG).



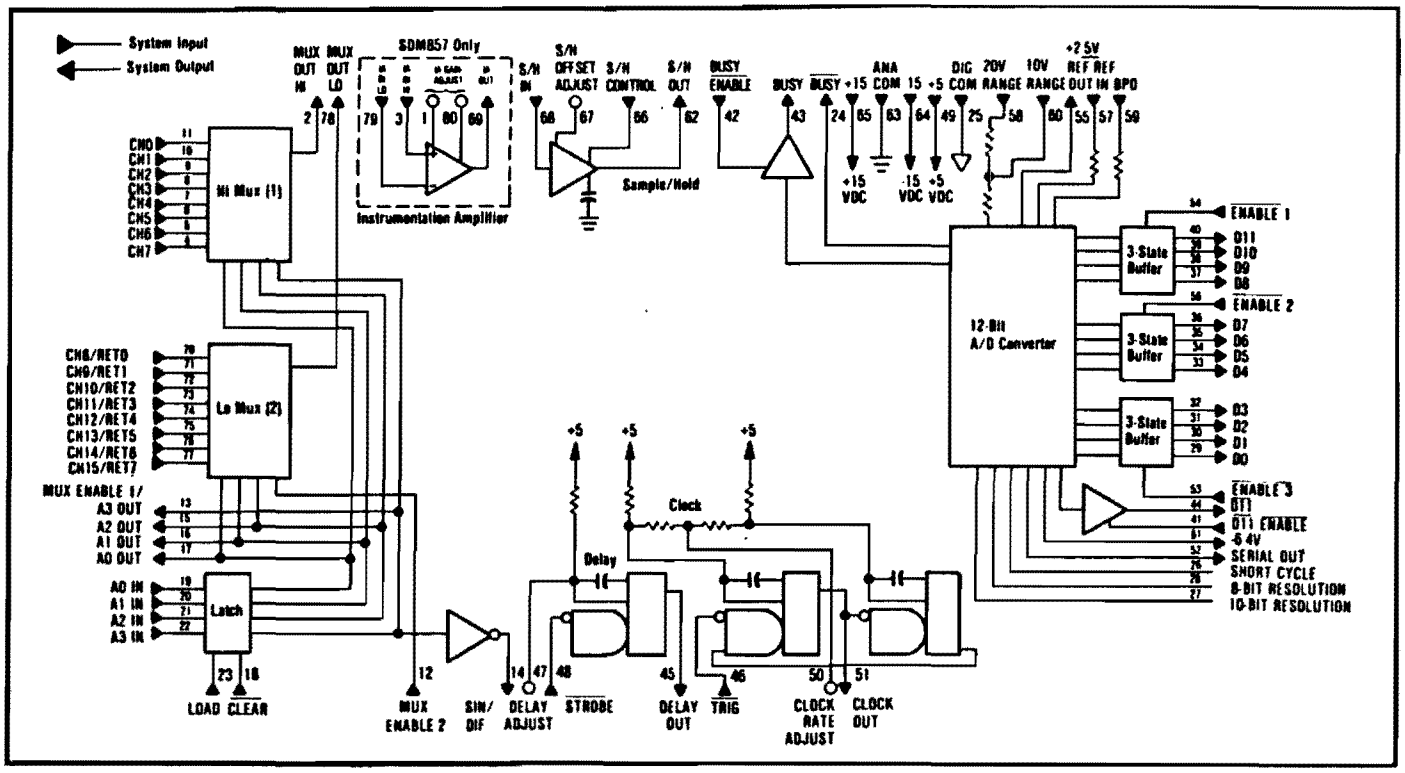


FIGURE 1. SDM856/857 Block Diagram.

SYSTEM PERFORMANCE

SDM856 and SDM857 are configured for random channel selection. With the addition of an external counter they can be configured to continuously sequence through all analog channels or sequence through all analog channels on command from an external trigger.

With the appropriate 4-bit (single-ended) or 3-bit (differential) channel address on the latch inputs, and DELAY OUT (pin 45) tied to the LOAD input (pin 23), a negative going edge is applied to the $\overline{\text{STROBE}}$ input (pin 48). This starts the delay timer, latches the multiplexer address, and allows the input signal to pass through the multiplexer, instrumentation amplifier and sample/hold and settle to its final value before starting the A/D conversion. The DELAY OUT signal (pin 45) is also connected to the $\overline{\text{TRIG}}$ input (pin 46) and the A/D conversion is initiated on the negative-going edge. The S/H CONTROL input (pin 66) is connected to $\overline{\text{BUSY}}$ (pin 24) so that the sample/hold is in the HOLD mode during the A/D conversion.

By using overlap programming the settling time effects of the analog multiplexer and instrumentation amplifier can be reduced, extending throughput sampling rates up to 29kHz for 12-bit and 70kHz for 8-bit resolution (ADC short-cycled). This mode of operation is most useful when converting low level inputs to accommodate the increased settling time of the instrumentation amplifier. Overlap programming is accomplished by connecting BUSY to $\overline{\text{STROBE}}$ and S/H CONTROL; DELAY OUT to LOAD and $\overline{\text{TRIG}}$. In this mode of operation the address of the next channel to be converted is latched and the output of the instrumentation amplifier allowed to settle to a new value during the present conversion.

DIGITAL INPUT SPECIFICATIONS

Address Inputs (A0 - A3)	One standard LSTTL load, positive true
Address Coding	4-bit binary
LOAD	One standard LSTTL load, positive true, address loaded on positive edge.
$\overline{\text{CLEAR}}$	One standard LSTTL load, negative true, low level clears address latch.
$\overline{\text{STROBE}}$	One standard LSTTL load, high-to-low transition triggers the delay timer.
$\overline{\text{TRIG}}$	One standard LSTTL load, a negative going edge initiates the A/D conversion.
SHORT CYCLE	One standard LSTTL load, logic 1 for 12-bit resolution. Connect to "8-bit" or "10-bit" for 8- or 10-bit resolution.
$\overline{\text{ENABLE 1}}$, $\overline{\text{ENABLE 2}}$, $\overline{\text{ENABLE 3}}$, $\overline{\text{DTI ENABLE}}$, BUSY $\overline{\text{ENABLE}}$	One standard LSTTL load, a low level enables the 3-state output.
S/H CONTROL	TTL compatible, 10μA maximum input current. Logic 0 = Hold mode. Logic 1 = Sample (track) mode.
MUX ENABLE 2	TTL compatible, 2μA input current. Logic 0 enables multiplexer 2 (channels 8-15).

DIGITAL OUTPUT SPECIFICATIONS

Parallel Data Outputs	5 standard TTL loads, positive true, 3-state.
Serial Output	2 standard TTL loads, positive true, NRZ, time serial data output beginning with DTI (see Timing Diagram)
$\overline{\text{DTI}}$	5 standard TTL loads, positive true, 3-state.
$\overline{\text{BUSY}}$	5 standard TTL loads, low during A/D conversion
BUSY	5 standard TTL loads, high during A/D conversion, 3-state
CLOCK OUT	5 standard TTL loads, for synchronizing serial out data (see Timing Diagram).
Address Outputs (A0 - A3)	5 LSTTL loads, positive true
DELAY OUT	5 standard TTL loads, high during delay period, triggered by Strobe input.
SIN/DIF	5 LSTTL loads, high while addressing channels 0-7, low while addressing channels 8-15.

Appendix V

BURR-BROWN®

UAF11
UAF21

UNIVERSAL ACTIVE FILTERS

FEATURES

- **SAVES DESIGN TIME**
  - User-tuneable frequency, Q-factor, gain
  - Calculate only three resistance values
  - Design directly from this data sheet
  - Completely characterized parameters
- **IMPROVED PERFORMANCE**
  - Wide frequency ranges
  - UAF11 - 0.001Hz to 20kHz
  - UAF21 - 0.001Hz to 200kHz
  - 1% frequency accuracy
  - Q range of 0.5 to 500
  - Reliable hybrid construction
  - NPO capacitors and thin-film resistors

APPLICATIONS

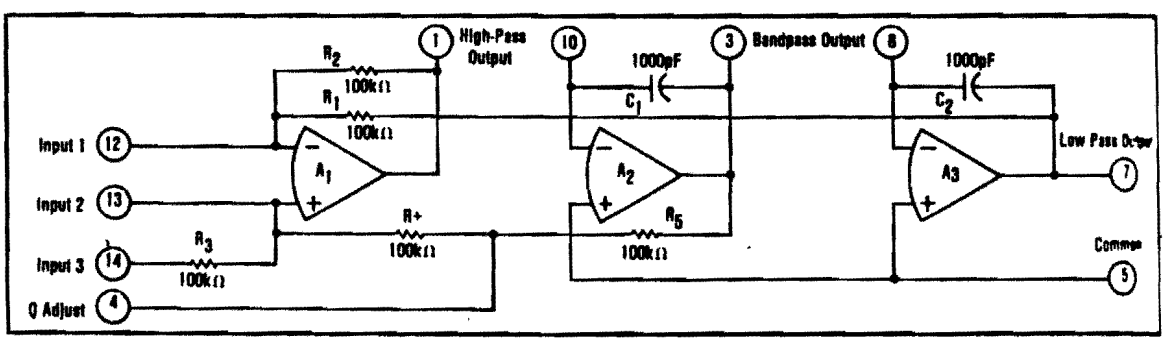
- **FILTER CONFIGURATIONS**
  - Butterworth
  - Bessel
  - Chebyschev
- **FILTER FUNCTIONS**
  - Low pass
  - High pass
  - Bandpass
  - Band reject

DESCRIPTION

The UAF11's and UAF21's are low cost universal active filters. These versatile units can easily be tailored to any active filter application using the extensive information provided in this data sheet. UAF's are excellent choices for use in communications equipment, test equipment (engine analyzers, aircraft and automotive test, medical test, etc.), servo systems, process control equipment, sonar and many others.

The UAF11's and UAF21's are complete two-pole active filters with the addition of four external resistors that provide the user easy control of the Q-factor, resonant frequency and gain. Any complex filter response can be obtained by cascading these units. Three separate outputs provide low-pass, high-pass, and bandpass transfer functions. A band-reject (notch) transfer function may be realized simply by summing the high-pass and low-pass outputs.

Since these UAF's are so versatile and flexible, they can be stocked by the user in quantity for use as building blocks whenever the requirement arises. This means instant availability and the UAF purchases may be made in volume to take advantage of quantity price discounts.



International Airport Industrial Park - P.O. Box 11400 - Tucson, Arizona 85734 - Tel. (602) 746-1111 - Twx: 910-952-1111 - Cable: BBRCORP - Telex: 66-6491

Appendix VI

ANALOG DEVICES

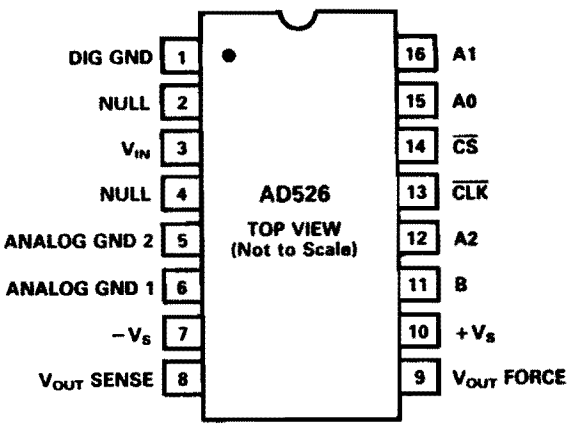
Software Programmable Gain Amplifier

AD526

FEATURES

- **Digitally Programmable Binary Gains from 1 to 16**
- **Two-Chip Cascade Mode Achieves Binary Gain from 1 to 256**
- **Gain Error:**
  - **0.01% max, Gain = 1, 2, 4 (C Grade)**
  - **0.02% max, Gain = 8, 16 (C Grade)**
  - **0.5ppm/°C Drift Over Temperature**
- **Fast Settling Time**
  - **10V Signal Change:**
    - **0.01% in 4.5μs (Gain = 16)**
  - **Gain Change:**
    - **0.01% in 5.6μs (Gain = 16)**
- **Low Nonlinearity: ±0.005% FSR max (J Grade)**
- **Excellent dc Accuracy:**
  - **Offset Voltage: 0.5mV max (C Grade)**
  - **Offset Voltage Drift: 3μV/°C (C Grade)**
- **TTL Compatible Digital Inputs**

AD526 PIN CONFIGURATION



PRODUCT DESCRIPTION

The AD526 is a single-ended, monolithic software programmable gain amplifier (SPGA) that provides gains of 1, 2, 4, 8 and 16. It is complete, including amplifier, resistor network and TTL-compatible latched inputs, and requires no external components.

Low gain error and low nonlinearity make the AD526 ideal for precision instrumentation applications requiring programmable gain. The small signal bandwidth is 350kHz at a gain of 16. In addition, the AD526 provides excellent dc precision. The FET-input stage results in a low bias current of 50pA. A guaranteed maximum input offset voltage of 0.5mV max (C grade) and low gain error (0.01%, G = 1, 2, 4, C grade) are accomplished using Analog Devices' laser trimming technology.

To provide flexibility to the system designer, the AD526 can be operated in either latched or transparent mode. The force/sense configuration preserves accuracy when the output is connected to remote or low impedance loads.

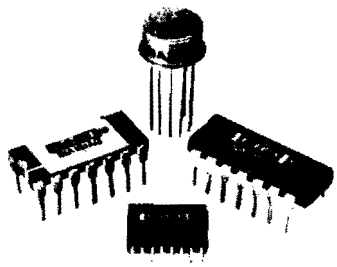
The AD526 is offered in one commercial (0 to +70°C) grade, J, and three industrial grades, A, B and C, which are specified from −40°C to +85°C. The S grade is specified from −55°C to +125°C. The military version is available processed to MIL-STD 883B, Rev C. The J grade is supplied in a 16-pin plastic DIP, and the other grades are offered in a 16-pin hermetic side-brazed ceramic DIP.

APPLICATION HIGHLIGHTS

1. **Dynamic Range Extension for ADC Systems:** A single AD526 in conjunction with a 12-bit ADC can provide 96dB of dynamic range for ADC systems.
2. **Gain Ranging Pre-Amps:** The AD526 offers complete digital gain control with precise gains in binary steps from 1 to 16. Additional gains of 32, 64, 128 and 256 are possible by cascading two AD526s.

Appendix VII

BURR-BROWN®



INA101

MILITARY & DIE VERSIONS AVAILABLE

Very-High Accuracy INSTRUMENTATION AMPLIFIER

FEATURES

- ULTRA-LOW VOLTAGE DRIFT - 0.25μV/°C
- LOW OFFSET VOLTAGE - 25μV
- LOW NONLINEARITY - 0.002%
- LOW NOISE - 13nV/$\sqrt{Hz}$ at f_0 = 1kHz
- HIGH CMR - 106dB at 60Hz
- HIGH INPUT IMPEDANCE - $10^{10}\Omega$
- LOW COST, TO-100, CERAMIC DIP AND PLASTIC PACKAGE

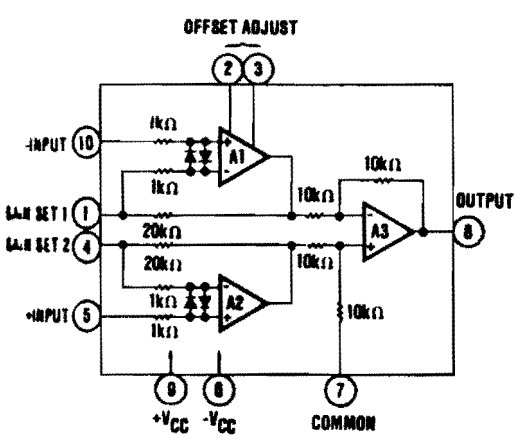
APPLICATIONS

- AMPLIFICATION OF SIGNALS FROM SOURCES SUCH AS:
  - Strain Gages
  - Thermocouples
  - RTDs
- REMOTE TRANSDUCERS
- LOW LEVEL SIGNALS
- MEDICAL INSTRUMENTATION

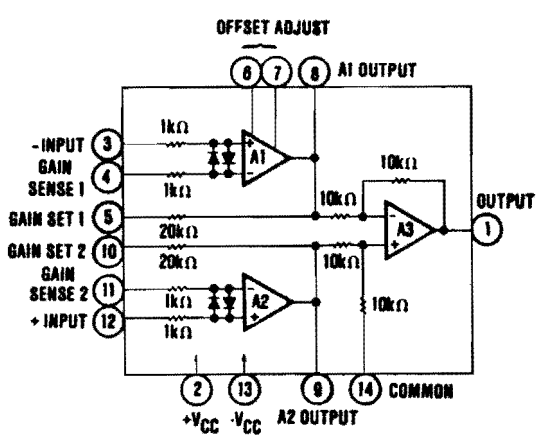
DESCRIPTION

The INA101 is a high accuracy, multistage, integrated-circuit instrumentation amplifier designed for signal conditioning requirements where very-high performance is desired. All circuits, including the interconnected laser-trimmed thin-film resistors, are integrated on a single monolithic substrate.

A multiamplifier design is used to provide the highest performance and maximum versatility with monolithic construction for low cost. The input stage uses Burr-Brown's ultra-low drift, low noise technology to provide exceptional input characteristics.



M Package

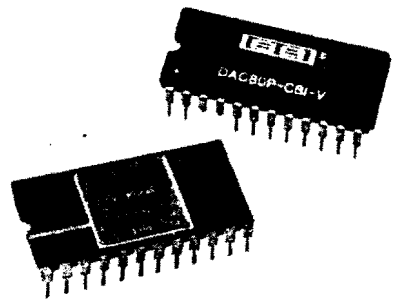


G and P Packages

International Airport Industrial Park - P.O. Box 11400 - Tucson, Arizona 85734 - Tel. (602) 746-1111 - Twx: 910-952-1111 - Cable: BBRCORP - Telex: 66-6491

Appendix VIII

BURR-BROWN®



DAC80
DAC80P

AVAILABLE IN DIE FORM

Monolithic 12-Bit DIGITAL-TO-ANALOG CONVERTERS

FEATURES

- INDUSTRY STANDARD PINOUT
- LOW POWER DISSIPATION: 345mW
- FULL ±10V SWING WITH V_{CC} = ±12VDC
- DIGITAL INPUTS ARE TTL- AND CMOS-COMPATIBLE
- GUARANTEED SPECIFICATIONS WITH ±12V AND ±15V SUPPLIES
- SINGLE-CHIP DESIGN
- ±1/2LSB MAXIMUM NONLINEARITY, 0°C to +70°C
- GUARANTEED MONOTONICITY, 0°C to +70°C
- TWO PACKAGE OPTIONS: Hermetic side-brazed ceramic and low-cost molded plastic
- SETTLING TIME: 4μs max to ±0.01% of Full Scale

DESCRIPTION

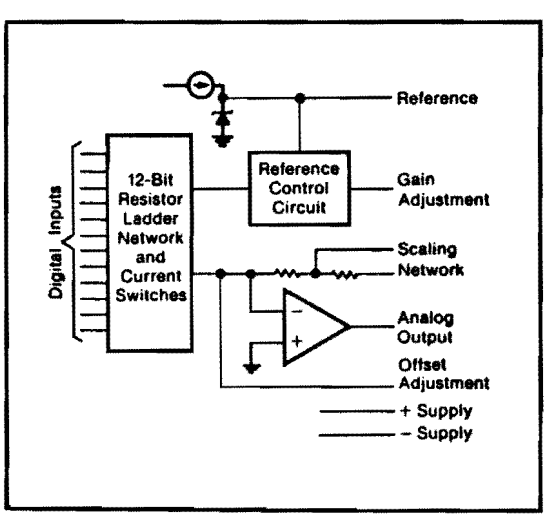
This monolithic digital-to-analog converter is pin-for-pin equivalent to the industry standard DAC80, first introduced by Burr-Brown. Its single-chip design includes the output amplifier and provides a highly stable reference capable of supplying up to 2.5mA to an external load without degradation of D/A performance.

This converter uses proven circuit techniques to provide accurate and reliable performance over temperature and power supply variations. The use of a buried zener diode as the basis for the internal reference contributes to the high stability and low noise of the device. Advanced methods of laser trimming result in precision output current and output amplifier feedback resistors, as well as low integral and differential linearity errors. Innovative circuit design enables the DAC80 to operate at supply voltages as low as ±11.4V with no loss in performance or accuracy over any range of output voltage. The lower power dissipation of this 118-mil by 121-mil chip results in higher reliability and greater long term stability.

Burr-Brown has further enhanced the reliability of the monolithic DAC80 by offering a hermetic, side-brazed, ceramic package. In addition, ease of use has been enhanced by eliminating the need for a +5V logic power supply.

For applications requiring both reliability and low cost, the DAC80P in a molded plastic package offers the same electrical performance over temperature as the ceramic model. The DAC80P is available with either voltage or current output.

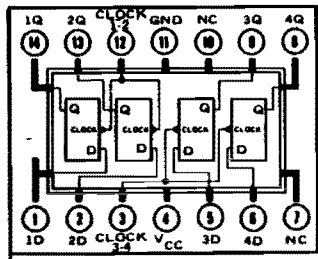
For designs that require a wider temperature range, see Burr-Brown models DAC85H and DAC87H. For designs that require complementary coded decimal inputs, see Burr-Brown model DAC80-CCD-V (-I).



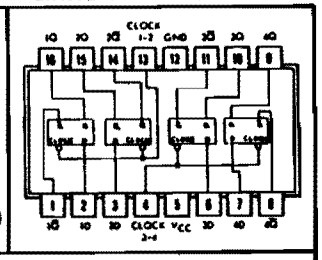
International Airport Industrial Park · P.O. Box 11400 · Tucson, Arizona 85734 · Tel. (602) 746-1111 · Twx: 910-952-1111 · Cable: BBRCORP · Telex: 66-6491

CIRCUIT TYPES SN5475, SN5477, SN54100, SN7475, SN7477, SN74100
8-BIT AND 4-BIT BISTABLE LATCHES

SN5477/SN7477
W FLAT PACKAGE
(TOP VIEW)



SN5475/SN7475
J OR N DUAL-IN-LINE
OR W FLAT PACKAGE (TOP VIEW)†



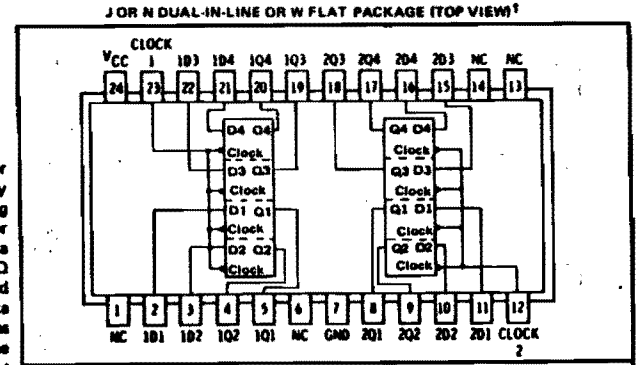
positive logic: see truth table

logic

TRUTH TABLE (Each Latch)	
t_n	t_{n+1}
D	Q
1	1
0	0

NOTES: 1. t_n = bit time before clock negative-going transition.
2. t_{n+1} = bit time after clock negative-going transition.
NC—No internal connection

SN54100/SN74100
J OR N DUAL-IN-LINE OR W FLAT PACKAGE (TOP VIEW)†


CLOCK VCC 1 1D3 1D4 1Q4 1Q3 2Q3 2Q4 2D4 2D3 NC NC
24 23 22 21 20 19 18 17 16 15 14 13
D4 Q4 Clock D3 Q3 Clock D1 Q1 Clock Q2 Q2 Clock
Q4 D4 Clock Q3 D3 Clock Q1 D1 Clock Q2 Q2 Clock
1 2 3 4 5 6 7 8 9 10 11 12
NC 1D1 1D2 1Q2 1Q1 NC GND 2Q1 2Q2 2D2 2D1 CLOCK 2

positive logic: see truth table

†Pin assignments for these circuits are the same for all packages

description

These latches are ideally suited for use as temporary storage for binary information between processing units and input/output or indicator units. Information present at a data (D) input is transferred to the Q output when the clock is high, and the Q output will follow the data input as long as the clock remains high. When the clock goes low, the information (that was present at the data input at the time the transition occurred) is retained at the Q output until the clock is permitted to go high.

The SN5475/SN7475 features complementary Q and $\overline{Q}$ outputs from a 4-bit latch, and is available in the 16-pin packages. For higher component density applications the SN5477/SN7477 4-bit latch is available in the 14-pin flat package. The SN54100/SN74100 features two independent quadruple latches in a single 24-pin dual-in-line package.

These circuits are completely compatible with all popular TTL or DTL families. Typical power dissipation is 40 milliwatts per latch. The Series 54 circuits are characterized for operation over the full military temperature range of −55°C to 125°C and Series 74 circuits are characterized for operation from 0°C to 70°C.

absolute maximum ratings (over operating temperature range unless otherwise noted)

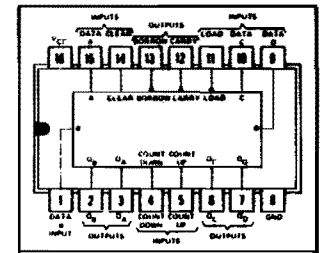
Supply Voltage, V_{CC} (See Note 3) . . . . . 7 V
Input Voltage, V_{in} (See Notes 3 and 4) . . . . . 5.5 V
Operating Free-Air Temperature Range: SN5475 Circuits . . . . . −55°C to 125°C
SN7475 Circuits . . . . . 0°C to 70°C
Storage Temperature Range . . . . . −65°C to 150°C

NOTES: 3. These voltage values are with respect to network ground terminal.
4. Input signals must be zero or positive with respect to network ground terminal.

CIRCUIT TYPES SN54192, SN54193, SN74192, SN74193
SYNCHRONOUS 4-BIT UP/DOWN COUNTERS (DUAL CLOCK WITH CLEAR)

- Cascading Circuitry Provided Internally
- Synchronous Operation
- Individual Preset to Each Flip-Flop
- Fully Independent Clear Input
- Typical Maximum Input Count Frequency . . . 32 MHz

J OR N
DUAL-IN-LINE PACKAGE (TOP VIEW)


INPUTS OUTPUTS INPUTS
VCC DATA A DATA B DATA C LOAD DATA D
16 15 14 13 12 11 10 9
A CLEAR BORROW CARRY LOAD C
QA QB COUNT DOWN COUNT UP QC QD
1 2 3 4 5 6 7 8
DATA A INPUT QA QB OUTPUTS COUNT DOWN COUNT UP INPUTS QC QD OUTPUTS GND

logic: Low input to load sets Q_A = A, Q_B = B, Q_C = C, and Q_D = D

description

These monolithic circuits are synchronous reversible (up/down) counters having a complexity of 55 equivalent gates. The SN54192 and SN74192 are BCD counters and the SN54193 and SN74193 are 4-bit binary counters. Synchronous operation is provided by having all flip-flops clocked simultaneously so that the outputs change coincidently with each other when so instructed by the steering logic. This mode of operation eliminates the output counting spikes which are normally associated with asynchronous (ripple-clock) counters.

The outputs of the four master-slave flip-flops are triggered by a low-to-high-level transition of either count (clock) input. The direction of counting is determined by which count input is pulsed while the other count input is high.

All four counters are fully programmable; that is, the outputs may be preset to any state by entering the desired data at the data inputs while the load input is low. The output will change to agree with the data inputs independently of the count pulses. This feature allows the counters to be used as modulo-N dividers by simply modifying the count length with the preset inputs.

A clear input has been provided which forces all outputs to the low level when a high level is applied. The clear function is independent of the count and load inputs. An input buffer has been placed on the clear, count, and load inputs to lower the drive requirements to one normalized Series 54/74 load. This is important when the output of the driving circuitry is somewhat limited.

These counters were designed to be cascaded without the need for external circuitry. Both borrow and carry outputs are available to cascade both the up- and down-counting functions. The borrow output produces a pulse equal in width to the count-down input when the counter underflows. Similarly, the carry output produces a pulse equal in width to the count-up input when an overflow condition exists. The counters can then be easily cascaded by feeding the borrow and carry outputs to the count-down and count-up inputs respectively of the succeeding counter.

Power dissipation is typically 325 milliwatts for either the decade or binary version. Maximum input count frequency is typically 32 megahertz and is guaranteed to be 25 MHz minimum. All inputs are buffered and represent only one normalized Series 54/74 load. Input clamping diodes are provided to minimize transmission-line effects and thereby simplify system design. The SN54192 and SN54193 are characterized for operation over the full military temperature range of −55°C to 125°C; the SN74192 and SN74193 are characterized for operation from 0°C to 70°C.

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage V_{CC} (see Note 1) . . . . . 7 V
Input voltage (see Note 1) . . . . . 5.5 V
Operating free-air temperature range: SN54192 and SN54193 Circuits . . . . . −55°C to 125°C
SN74192 and SN74193 Circuits . . . . . 0°C to 70°C
Storage temperature range . . . . . −65°C to 150°C

NOTE 1: Voltage values are with respect to network ground terminal.

Appendix X

Power amplifier

Axodyn-Power Servo Amplifier 05 LV 05

B.B.C. : Brown Bovery company

Table 1 Summary of Specification

Axodyn power servo amplifier		Power supplies		Output						
Type	Order No. GJV 160...	Supply Voltage	Power Consumption	d. c. voltage rating①	d. c. current rating②	Dynam-peak current③	Output impedance	Power rating④	Dynam-peak power	weight
		V	VA	V-	A-	A-	Ω	W	W	kg
05 LV 01...-E	...1001 R1	220 V/1~	200	±13	± 8	±16	0,30	100	210	6,0
05 LV 02...-E	...1002 R1	220 V/1~	250	±24	± 6	±15	0,20	145	360	8,5
05 LV 03...-E	...1003 R1	220 V/1~	550	±52	± 6	±15	0,30	360	780	18,0
05 LV 04...-E	...1004 R1	380 V/3~	1200	+80/-12	±10	+30/-20	0,30	800	2400	37,0
05 LV 05...-E	...1005 R1	380 V/3~	1200	±45	±20	±40	0,30	900	1800	37,0
05 LV 06...-E	...1006 R1	220 V/1~	500	±40	±10	±20	0,30	400	800	18,0
05 LV 07...-E	...1007 R1	220 V/1~	400	±24	±10	±20	0,20	240	480	10,0
05 LV 08...-E	...1008 R1	220 V/1~	550	70/-12	± 6	+15/-10	0,30	420	1050	18,0
05 LV 09...-E	...1009 R1	220 V/1~	550	±24	±15	±30	0,25	360	720	18,0

① at rated d. c. current

② can be adjusted from 10% to 100% of rated d. c. current

③ max 2 sec. decay time constant can be set

④ reduction 2%/°C from 40°C ambient temperature increase

Table 2

Regulating and Control Section

Operating ranges

Tolerance of connection voltage	+ 10% -10%
Frequency	50...60 Hz
Guaranted temperature range	0°C...40°C
Operational temperature range	from -10°C...+60°C

Stabilised current supply

positive output voltage	+14.1 V...+15.9 V (specimen scatter)
negative output voltage	-14.1 V...-15.9 V
Temperature drift	< 2.25 mV/°C
max. additional loading	± 20 mA

Speed regulator

Rated voltage	± 15 V
Actual voltage	± 15 V
max. ext. rated voltage	± 30 V
Input resistance rated value	30 kΩ
Large signal gain	200 000
In-phase suppression	90 dB
Limiting frequency (without network)	10 kHz
Offset voltage adjustment	by potentiometer

Tacho control

Adjustment range	0...100%
Control range	1 : 5000
Control error	± 0.5%

IR Compensation

Actuating range	10...100%
Control range	1 : 100
Control error	± 5%

Current limiting

Current limiting (continuous current)	10...100%
Dynamic peak current	see table 1
Max. duration of dynamic peak current	see Fig. 11

Note

The limits of the control range are associated with the permissible control error. Indication of errors is related to rated output voltage (speed) and the following interference variables; a wider control range is possible with a larger error:

Change in load	4 : 1
Mains voltage fluctuation	± 10%
Temperature change	± 5%

CEM Cie Electro-Mécanique
BBC BROWN BOVERI
BROWN BOVERI NEDERLAND B.V.
Elektroweg 22
Postbus 301
Rotterdam Tel:010-180 280

Ets PARVEX 27-29, rue Lucien Juy, 21007 - DIJON-CEDEX - FRANCE

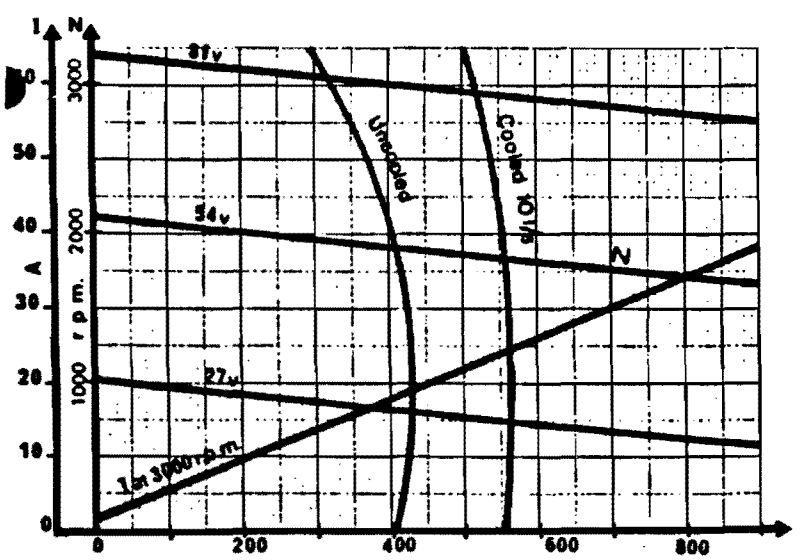
DÉPt VITESSE VARIABLE **Phone (80) 41.81.18 — Telex 350653**

	SYMBOLS		MC 19 P Uncooled	MC 19 P Cooled (2) 10 liter/sec
1 — MOTOR RATINGS (1)				
11. Rated Torque	Cn	cm.N	320	510
12. Rated Speed	Nn	r.p.m.	3000	3000
13. Rated Output	Pn	W	1000	1600
14. Rated Voltage	Un	V	83	87
15. Rated Current	In	A	14,4	22,2
16. Maximum at Very Low Speed *	Icc	A	16,5	22,3
17. Pulse Torque (intermittent operation) (3)	Cimp	cm.N	2440	2440
18. Maximum Speed with no external load **	Nmax	r.p.m.	5000	5000
2 — MOTOR CONSTANTS				
21. E.M.F./1000 RPM	K_E	V	25,5	25,5
22. Torque Constant/Ampere	K_T	cm.N	24,4	24,4
23. Regulation Constant Voltage/cm.N	K_N	r.p.m.	0,3	0,3
24. Friction Torque	T_f	cm.N	10	10
25. Damping Constant/1000 RPM	K_D	cm.N	8	8
26. Terminal resistance (4)	R	Ω	0,46	0,46
27. Armature Inductance	L	μH	<100	<100
28. Total Inertia	J	g. cm2	12000	12000
29. Mechanical Time Constant	τ	ms	9,2	9,2
2.10 Power Rate (5)	Ps	KW/s	500	500

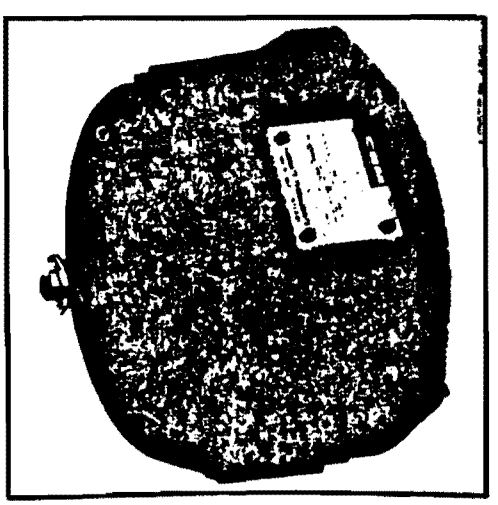
* Stall motor : ask us for max. current
** For other duty cycles : ask us

★ 4 parallel layers

3 — PERFORMANCE CHARACTERISTICS, CONTINUOUS OPERATION



Torque (cm. N)



Data given are in practice independent of the armature temperature

1) Mounting on metal plate with thermal insulation (400 x 400 x 10 mm), and pure DC supply. Ambient temperature 0° C to 40° C

2) Loss in motor : 13 mm of H2O

3) Allowed cycle S3, 50 ms, 1%

4) Value included rotor plus contact resistance which does not vary with armature temperature since armature resistance increases when contact resistance decreases

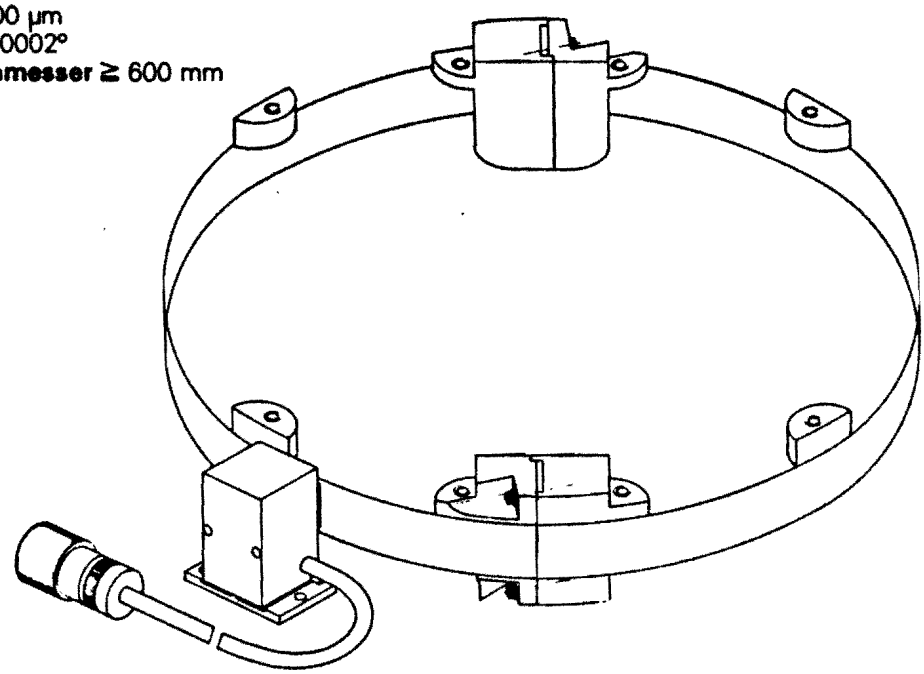
5) Calculated from the formula : $\frac{(\text{Pulse torque})^2}{(\quad)}$

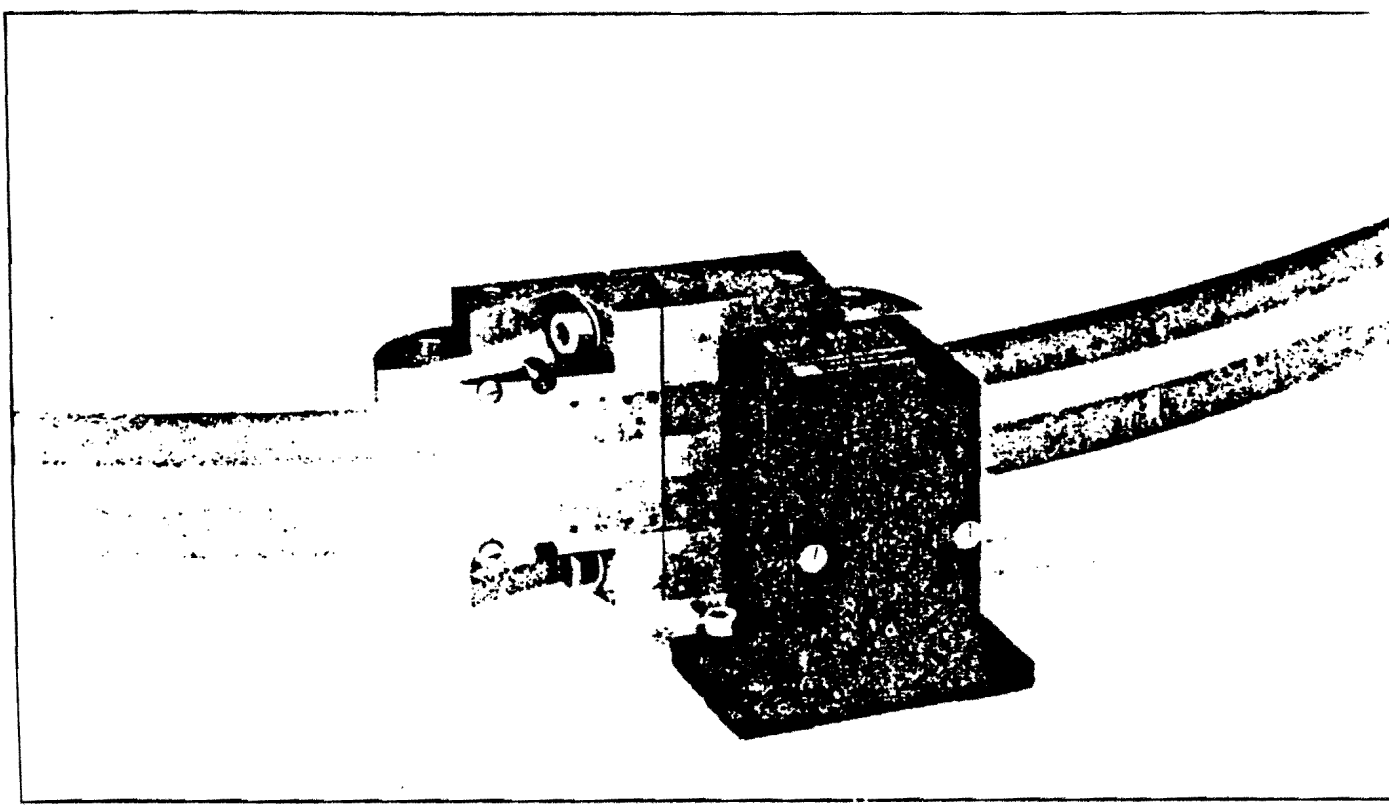
HEIDENHAIN

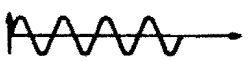
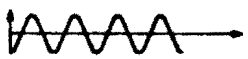
Winkelmeßgerät

LIDA 36(

Inkrementales offenes Winkelmeßgerät
mit AURODUR-Stahlband
Teilungsperiode 100 µm
Meßschritt bis 0,00002°
Bandauflagedurchmesser ≥ 600 mm





Strichzahlen [z]	abhängig vom Bandauflagedurchmesser (D), da Teilungsperiode des AURODUR Stahl-Maßbandes immer **100 µm** $z = \text{Int}\left[\frac{(Dmm + 0{,}3\ mm) \cdot \pi}{0{,}1\ mm}\right]$ Int: Ganzzahl-Anteil des in Klammern stehenden Ausdrucks
Mechanische Kennwerte	
Bandauflagedurchmesser	$\geq 600\ mm_{k7}$
Maßband-Teilstücke	≤ 3000 mm Stahl-Maßband-Teilstücke können über Spannschlösser miteinander verbunden werden.
Wärmeausdehnungskoeffizient des AURODUR-Stahlbandes	$10{,}5 \cdot 10^{-6}\ K^{-1}$
Referenzmarken	In der Mitte (Toleranzbereich ± 10 mm) des Stahl-Maßbandteilstückes und dann ausgehend im 50-mm-Raster. Damit die erste und letzte Referenzmarke nicht näher als 5 mm am Stahl-Maßband-Stoß liegt, wird der Toleranzbereich von ± 10 mm ausgenutzt.
Maßband-Genauigkeitsklasse	± 5 µm
Größter Unterteilungsfehler	± 2 µm (± 1 µm nur mit EXE 700 nach Abgleich) ± 5 µm an den Stoßstellen
zulässige Beschleunigung	$100\ m/s^2$
zulässige Stoßbelastung	$200\ m/s^2$, 5 ms
Schutzart	Abtasteinheit staub- und spritzwassergeschützt nach IP54 (DIN 40050)
Korrosionsschutz	Abtasteinheit: eloxiertes Aluminium Stahl-Maßband, Spannschlösser und Endspannstücke: rostfreier Stahl
Arbeitstemperatur-Bereich	Abtasteinheit: 0° C bis 45° C Stahl-Maßband und Maßbandträger: 0° C bis 50° C Nur wenn der Maßbandträger aus einem Material besteht, dessen Wärmeausdehnungskoeffizient zwischen $9 \cdot 10^{-6}$ und $12 \cdot 10^{-6}\ K^{-1}$ (z.B. Gußeisen oder ferritischer Stahl) beträgt. Bei höheren Wärmeausdehnungskoeffizienten (z.B. Aluminium) gilt ein eingeschränkter Temperaturbereich von 10° C bis 30° C.
Lagertemperatur-Bereich	Abtasteinheit: −30° C bis 70° C Stahl-Maßband und Maßbandträger: 0° C bis 50° C
relative Feuchte	max. 80%
Gewicht	Abtasteinheit 350 g Spannschloß 300 g Endspannstück 300 g Stahl-Maßband 60 g/m
Elektrische Kennwerte	
Lichtquelle	LED mit Vorwiderstand: 5 V ± 10%, < 120 mA
Impulsformer-Elektronik	a) in Zähler eingebaut b) extern, siehe EXE-Druckschrift
Ausgangssignale Inkremental-Signale	I_{e1}  2 annähernd sinusförmige Signale I_{e1} und I_{e2} I_{e2} 
Referenz-Signal	I_{e0} 1 Signal I_{e0} pro Umdrehung
Signalgröße	I_{e1} ca. 11 μA_{ss} I_{e2} ca. 11 μA_{ss} I_{e0} ca. 5,5 µA* * Nutzanteil bei Last 1 kOhm
Höchste zulässige Drehzahl n_{max}	$n_{max}\ [min^{-1}] = \frac{f_{max}\ [kHz]}{z} \cdot 10^3 \cdot 60$ z: Anzahl der Teilungsperioden von 100 µm auf dem Umfang (Strichzahl) f_{max}: maximale Eingangsfrequenz der externen Impulsformer-Elektronik
zulässige Kabellänge zur Folgeelektronik	20 m (HEIDENHAIN-Kabel [3 (2 x 0,14) + 2 x 0,5] mm^2)

Appendix XI

```
MODULE PRINCIPLE;

(* This module is the heart of the program, it includes all the      *)
(* routines to run the acquisition and the calculation of the        *)
(* constraint. This module calls also procedures from other modules.*)
(* This program doesn't do the adjustment of offset, at the end of  *)
(* module a flowchart is given in this way.                          *)

$INCLUDE (:F4:INTER1.SRC)


(* The inclusion of INTER1.SRC allows this module to use the         *)
(* variables declared in INTER1.SRC and to call the other            *)
(* procedures and functions declared as public.                      *)

$INTERRUPT (TEACH_INT=0,REPLAY_INT=1,SAMPLE_INT=2,TEST_INT=5)
$INTERRUPT(DATA_READY=6, OVERRANGE=7)


PROGRAM PRINCIPLE;

   CONST DELAY=10;

   VAR I,J: INTEGER;


  (*****************************************************************)

  PROCEDURE ASSIGN_INTERRUPTS;

  (* This procedure atributes the level of the different interrupts *)
  (* used, it also permits to define the interrupt vectors.         *)


    BEGIN

       DISABLEINTERRUPTS; (*Disable the interrupts at the CPU level.*)

       SETINTERRUPT(00CH,TEACH_INT);
       SETINTERRUPT(001H,REPLAY_INT);
       SETINTERRUPT(002H,SAMPLE_INT);
       SETINTERRUPT(005H,TEST_INT);
       SETINTERRUPT(006H,DATA_READY);
       SETINTERRUPT(007H,OVERRANGE);


       OUTBYT(ADRR_MASK,00H);(*Reset the interrupt mask of the 8259.*)


       ENABLEINTERRUPTS; (* Enable the interrupts at the CPU level. *)


    END;
```

```
{* The procedures following are called when the hardware          *}
{* interrupts connected to the PIC are activated.                 *}


PROCEDURE TEACH_INT;{*This procedure services hardware interrupt 0.*}

  BEGIN
    PRINT30('Teach Interrupt Activated     ');
    TEACH:=TRUE;
  END;

PROCEDURE REPLAY_INT;{*This procedure services hardware interrupt 1*}

  BEGIN
    PRINT30('Replay Interrupt activated    ');
    REPLAY:=TRUE;
  END;

PROCEDURE SAMPLE_INT;{*This procedure services hardware interrupt 2*}

  BEGIN
    PRINT30('Sample Interrupt Activated    ');
    SAMPLE:=TRUE;
  END;

PROCEDURE TEST_INT;{*This procedure services hardware interrupt 5. *}

  BEGIN
    PRINT30('Test Interrupt Activated      ');
    TEST1:=TRUE;
  END;

PROCEDURE DATA_READY;{*This procedure services hardware interrupt 6*}

  BEGIN
    DATA:=TRUE;
  END;

PROCEDURE OVERRANGE;{*This procedure services harware interrupt 7. *}
  BEGIN
    CO('O');CO('V');CO('E');CO('R');
    OVER:=TRUE;
  END;


{*****************************************************************}
```

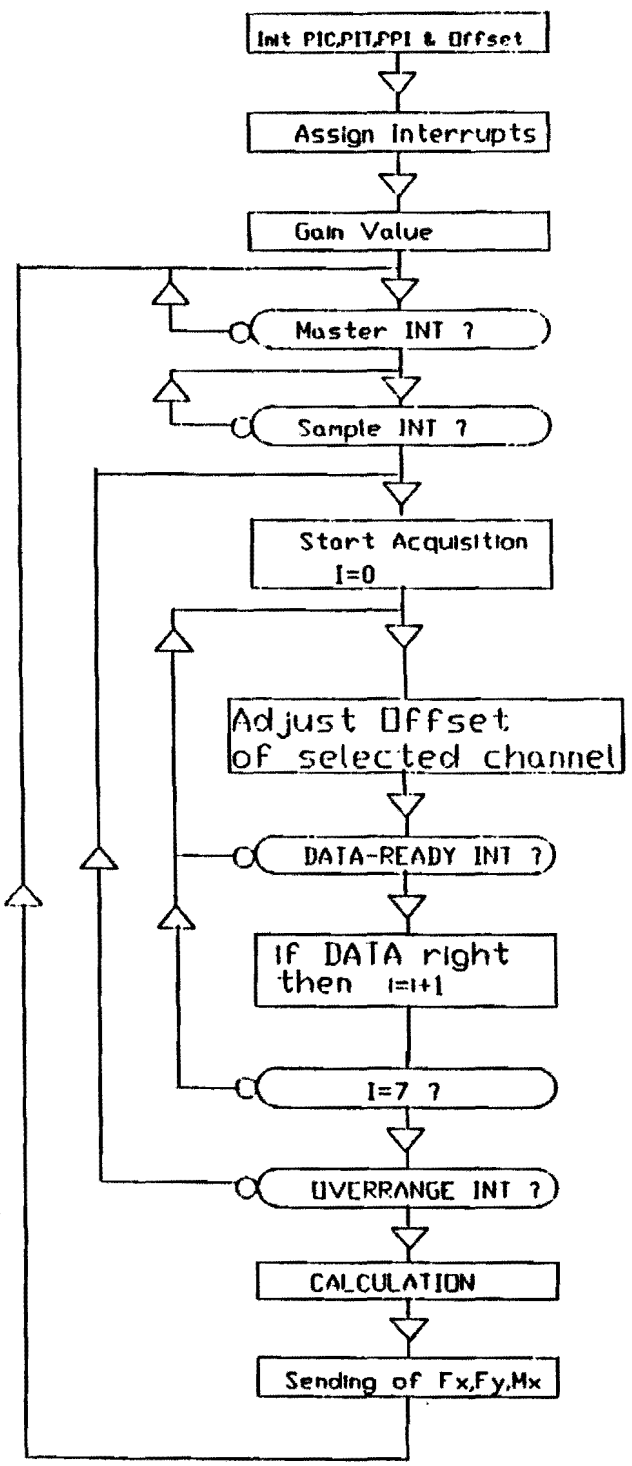
```
{*****************************************************************}

PROCEDURE CALCUL_VOLTAGE;

{* The TEMP_VOLTAGE matrix calculated in the main program is      *}
{* used to determine the VOLTAGE matrix.                          *}

   VAR          I,J,MAX_COEFF,OLDVAL : INTEGER;

   BEGIN
     MAX_COEFF:=2048;  {* 2048=2 P(11)                            *}
     OLDVAL:=0;     {* Initialization of this temporary variable.  *}
     FOR I:=1 TO 8  DO
       BEGIN
         IF TMP_VOLT[I,1]=1 THEN SIGN:=-1 ELSE SIGN:=1;
         {* The MSB defines the sign of the integer.             *}

         FOR J:=2 TO 12 DO
           BEGIN
            VOLTAGE_1[I]:=MAX_COEFF*TMP_VOLT[I,J] + OLDVAL;
            MAX_COEFF:=MAX_COEFF DIV 2;{*Get the other power of 2*}
            OLDVAL:=VOLTAGE_1[I];
           END;
        VOLTAGE_1[I]:=SIGN*VOLTAGE_1[I];
       END;

   END;

{*****************************************************************}

{*****************************************************************}


  PROCEDURE LINEAR;

  {* This procedure linearises the value of the voltage which is  *}
  {* supposed to keep a linear variation during the calculation   *}
  {* Time.                                                        *}

    VAR   I: INTEGER;

 BEGIN
  FOR I:=1 TO 8 DO
  BEGIN
   T1:=(8 - I)*DELTA + CALCULATION_TIME;
   T2:=CALCULATION_TIME + (7*DELTA);
   VOLTAGE[I]:=ROUND(VOLTAGE_1[I] + ((VOLTAGE_1[I] - VOLTAGE_0[I] *T1)/T2)
   VOLTAGE_0[I]:=VOLTAGE_1[I];
  END;
 END;

{*****************************************************************}
```

```
{*************************************************************************}

PROCEDURE MULT_MATRICE;

{* The multiplication of the VOLTAGE matrix by the transformation *}
{* matrix leads to the constraint matrix.                         *}

   VAR I: INTEGER;

   BEGIN
     FX:=0;FY:=0;M:=0;   {* Initialisation of the variables.      *}

     FOR I:=1 TO 8 DO
       BEGIN
         FX:=VOLTAGE[I]*TM[I,1]+FX;
         FY:=VOLTAGE[I]*TM[I,2]+FY;
         M:=VOLTAGE[I]*TM[I,3]+M;
       END;

   END;

{*************************************************************************}

{*************************************************************************}

PROCEDURE SEND_DATA;

{* The sending of the data into the RAM board is easier with the  *}
{* MBWI (Multibus write integer) defined in the MBUS.LIB file.    *}

   BEGIN

     MBWI(0A0000H,FX); {* First data is the address pointed and    *}
     MBWI(0A0002H,FY); {* the second one is the data loaded.       *}
     MBWI(0A0004H,M);

   END;

{*************************************************************************}

{*************************************************************************}

PROCEDURE START_ACQUISITION;

{* The beginning of the acquisition is activated by the counter 0 *}
{* and the selection of the other channels by the use of counter 1*}

   VAR I: INTEGER;

   BEGIN

     OUTBYT(ADRR_TIMER_0,DATA_COUNTER_0);
     OUTBYT(ADRR_TIMER_1,DATA_COUNTER_1);

   END;

{*************************************************************************}
```

```
{*************************************************************************}

PROCEDURE CHECK_DATA;

{* This procedure issued the calcul data which is the present      *}
{* selected channel address. Moreover, the PPI binary inputs are   *}
{* tranformed in integers in order to be computed.                 *}

  VAR
      I: INTEGER;

  BEGIN
   FOR I:=1 TO 7 DO
    BEGIN
     ADRR[I]:=TMP_VOLT[I,16]+(TMP_VOLT[I,15]*2)+(TMP_VOLT[I,14]*4);
     IF ADRR[I]<>I THEN RIGHT:=FALSE;
    END;
     ADRR[8]:=TMP_VOLT[8,16]+(TMP_VOLT[8,15]*2)+(TMP_VOLT[I,14]*4);
     IF ADRR[8]<>0 THEN RIGHT:=FALSE;
  END;

{*************************************************************************}


{*************************************************************************}

BEGIN {* Start of the program.                                          *}

   CLS;LINE(5);TAB(10);
   INITIALISATION;    {* Initialisation of the peripheral devices. *}
   PRINT30('INITIALISATION  PPI/PIT/PIC    ');

   CLS;LINE(5);TAB(10);
   ASSIGN_INTERRUPTS; {* The interrupts are now enabled.          *}
   PRINT30('INTERRUPTS ASSIGNED            ');

   TEST1:=FALSE;REPLAY:=FALSE;TEACH:=FALSE;SAMPLE:=FALSE;
   {*All the interrupt flags have been reseted to be sure of them.*}

   INIT_TRANSF_MATRIX;

   CLS;LINE(10);TAB(10);
   RANGE;   {* Allows the user to select his amplification.        *}

   WHILE TRUE DO

       BEGIN
       LINE(1);TAB(5);
       PRINT30('WAITING FOR AN INTERRUPT       ');


     IF TEACH OR REPLAY OR TEST1 OR SAMPLE  THEN

         BEGIN

           DATA:=FALSE; {* Reset of the interface interrupt flag.*}
```

```
        REPEAT

           START_ACQUISITION;
           OVER:=FALSE;

           I:=0;
           CO('W');

            REPEAT

              IF DATA THEN
               BEGIN
                DATA:=FALSE;
                FOR J:=1 TO 8 DO
                  BEGIN
                    INBYT(ADRR_PORT_A,DATA_A[J]);
                    INBYT(ADRR_PORT_C,DATA_C[J]);
                  END;

           FOR J:=1 TO 8 DO
           BEGIN
            IF DATA_A[J] THEN TMP_VOLT[I,J]:=1 ELSE TMP_VOLT[I,J]:=0;
            IF DATA_C[J] THEN TMP_VOLT[I,J+8]:=1 ELSE TMP_VOLT[I,J+8]:=0;
           END;

                I:=I+1;
               END;

            UNTIL I=8;

            CHECK_DATA;


        UNTIL RIGHT AND NOT OVER;

        CALCUL_VOLTAGE;
        LINEAR;
        MULT_MATRICE;
        SEND_DATA;
        PRINT30('DATA SENT TO THE RAM BOARD !  ');

      END;

   END;

END.
```

Flowchart with the offset adjustment



Appendix XII

```
PUBLIC PRINCIPLE;


  CONST
            ADRR_INIT_PPI=     0CEH;
            ADRR_PORT_A =      0CBH;
            ADRR_PORT_B =      0CAH;
            ADRR_PORT_C =      0CCH;

            ADRR_ICW1 =        0C0H;
            ADRR_ICW2 =        0C2H;
            ADRR_ICW4 =        0C2H;
            ADRR_MASK =        0C2H;

            ADRR_INIT_PIT =    0D6H;
            ADRR_TIMER_0 =     0D0H;
            ADRR_TIMER_1 =     0D2H;

            DATA_INIT_PPI =    099H;

            DATA_ICW1 =        011H;
            DATA_ICW2 =        000H;
            DATA_ICW4 =        00FH;
            DATA_MASK =        0FFH;

            DATA_TIMER_0     = 01EH;
            DATA_TIMER_1     = 050H;
            DATA_COUNTER_0   = 00DH;
            DATA_COUNTER_1   = 00BH;
            CALCULATION_TIME=  170;
            DELTA            =  17;


TYPE
            STRING30 =                PACKED ARRAY [1..30] OF CHAR;
            MATRIX_8_B =            PACKED ARRAY [1..8] OF BOOLEAN;
            MATRIX_16_I =          PACKED ARRAY [1..16] OF INTEGER;
            MATRIX_8_16_I = PACKED ARRAY [1..8,1..16] OF INTEGER;
            MATRIX_8_I =            PACKED ARRAY [1..8] OF INTEGER;
            MATRIX_8_3_I =    PACKED ARRAY [1..8,1..3] OF INTEGER;

VAR
            RANGE_ACQUISITION:   INTEGER;
            SIGN,CALCUL:         INTEGER;
            FX,FY,M,T1,T2:       INTEGER;
```

```
        DATA_A,DATA_C:    MATRIX_8_8;
        TMP_VOLT:      MATRIX_8_16_I;
        VOLTAGE:          MATRIX_8_I;
        VOLTAGE_0,ADRR:   MATRIX_8_I;
        VOLTAGE_1:        MATRIX_8_I;
        TM:             MATRIX_8_3_I;
        RIGHT:                BOOLEAN;
        ANSWER:                  CHAR;
        REPLAY,TEACH,TEST1:   BOOLEAN;
        DATA,OVER,SAMPLE:     BOOLEAN;



PUBLIC HARDWARE;

        PROCEDURE INITIALISATION;
        PROCEDURE RANGE;
        PROCEDURE INIT_TRANSF_MATRIX;



PUBLIC SBCIOS;

        PROCEDURE CO(X:CHAR);
        FUNCTION CI:CHAR;


PUBLIC CONSOLE;

        PROCEDURE LINE(N:INTEGER);
        PROCEDURE TAB(LENGTH:INTEGER);
        PROCEDURE PRINT30(WORD30:STRING30);
        PROCEDURE CLS;


PUBLIC MBUS86;

       PROCEDURE MBWI(ADDRES:LONGINT;DONNE:INTEGER);
       PROCEDURE MBRI(ADDRES:LONGINT;DONNE:INTEGER);
```

Appendix XIII

```
MODULE HARDWARE;

$INCLUDE(:F4:INTER1.SRC)

PRIVATE HARDWARE;

  PROCEDURE INITIALISATION;

  {* This procedure initialised the PPI,PIT and PIC peripherals. It *}
  {* masks the interrupts of the PIC with the DATA mask which is    *}
  {* to FFH.                                                        *}

     BEGIN

       OUTBYT(ADRR_INIT_PPI,DATA_INIT_PPI);

       OUTBYT(ADRR_INIT_PIT,DATA_TIMER_0);
       OUTBYT(ADRR_INIT_PIT,DATA_TIMER_1);

       OUTBYT(ADRR_ICW1,DATA_ICW1);
       OUTBYT(ADRR_ICW2,DATA_ICW2);
       OUTBYT(ADRR_ICW4,DATA_ICW4);

       OUTBYT(ADRR_MASK,DATA_MASK);

     END;


  PROCEDURE RANGE;

  {* This procedure alows the user to select the gain of the        *}
  {* programmable amplifier.                                        *}

    BEGIN

      REPEAT

        CLS; LINE(10);TAB(10);
        PRINT30('RANGE OF AMPLIFICATION 1/2/4 ?');
        ANSWER:=CI; CO(ANSWER);

      UNTIL (ANSWER='1') OR (ANSWER='2') OR (ANSWER='4');

      CASE ANSWER OF

        '1': RANGE_ACQUISITION:=01H;
        '2': RANGE_ACQUISITION:=02H;
        '4': RANGE_ACQUISITION:=04H;
       END;
```

```
    OUTBYT(ADRR_PORT_B,RANGE_ACQUISITION);

  END;


PROCEDURE INIT_VOLTAGE_0;

(* This procedure initialises the voltage_0 to linearise the data *)
(* we suppose that at the beginning the voltage matrix is null.   *)

 VAR I:INTEGER;

 BEGIN
   FOR I:=1 TO 8 DO
     BEGIN
        VOLTAGE_0[I]:=0;
     END;
 END;



PROCEDURE INIT_TRANSF_MATRIX;

(* This procedure initialises the transformation matrix.          *)

  BEGIN

    TM[1,1]:=545;TM[2,1]:=14;TM[3,1]:=-567;TM[4,1]:=8;
    TM[5,1]:=525;TM[6,1]:=-7;TM[7,1]:=-537;TM[8,1]:=-8;

    TM[1,2]:=-3;TM[2,2]:=1385;TM[3,2]:=5;TM[4,2]:=-13;
    TM[5,2]:=1;TM[6,2]:=-1498;TM[7,2]:=2;TM[8,2]:=-97;

    TM[1,3]:=0;TM[2,3]:=-14;TM[3,3]:=0;TM[4,3]:=14;
    TM[5,3]:=0;TM[6,3]:=-15;TM[7,3]:=0;TM[8,3]:=14;

  END;
```

Appendix XIV

```
MODULE CONSOLE;

$NOLIST
$INCLUDE(:F4:INTER1.SRC)

PRIVATE CONSOLE;

     PROCEDURE LINE(N:INTEGER);

     (* This procedure permits to select the line to be printed.    *)

        VAR I:INTEGER;

        BEGIN

           FOR I:=1 TO N DO
               BEGIN
                 CO(CR);
                 CO(LF);
               END;

        END;


     PROCEDURE TAB(LENGTH:INTEGER);

     (* This instruction is used to move the cursor on a line       *)

        VAR I:INTEGER;

        BEGIN

          FOR I:=1 TO LENGTH DO CO(' ');

        END;


     PROCEDURE CLS;

        BEGIN

          LINE(25);

        END;


     PROCEDURE PRINT30(WORD30:STRING30);

     (* With this procedure, a word of 30 Characters,Blanks included *)
     (* can be printed on the screen.                                *)

        VAR I:INTEGER;

        BEGIN

          FOR I:=1 TO 30 DO CO(WORD30[I]);

        END;
```

Appendix XV

```
SERIES-III 8086 LOCATER, V2.5

INPUT FILE: :F4:MAIN.LNK
OUTPUT FILE: :F4:MAIN.STA
CONTROLS SPECIFIED IN INVOCATION COMMAND:
   TO :F4:MAIN.STA INITCODE(1000H) ORDER(CLASSES(CODE,
  DATA,STACK)) ADDRESSES(CLASSES(CODE(1400H),DATA(2400H),STACK(3400H)))
DATE:  22/02/88  TIME:


SYMBOL TABLE OF MODULE PRINCIPLE

BASE     OFFSET TYPE SYMBOL                    BASE     OFFSET TYPE SYMBOL

0140H    0485H  PUB  PRINCIPLE                 0240H    003CH  PUB  ADRR
0240H    017EH  PUB  ANSWER                    0240H    0172H  PUB  CALCUL
0240H    017AH  PUB  DATA                      0240H    0164H  PUB  DATA_A
0240H    015CH  PUB  DATA_C                    0240H    0170H  PUB  FX
0240H    016EH  PUB  FY                        0240H    016CH  PUB  M
0240H    0179H  PUB  OVER                      0240H    0176H  PUB  RANGE_ACQUISITIO
                                                                  -N
0240H    017DH  PUB  REPLAY                    0240H    017FH  PUB  RIGHT
0240H    0178H  PUB  SAMPLE                    0240H    0174H  PUB  SIGN
0240H    017CH  PUB  TEACH                     0240H    017BH  PUB  TEST1
0240H    000CH  PUB  TM                        0240H    005CH  PUB  TMP_VOLT
0240H    004CH  PUB  VOLTAGE                   01A8H    007CH  PUB  CLS
01A8H    000EH  PUB  LINE                      01A8H    0094H  PUB  PRINT30
01A8H    0049H  PUB  TAB                       01B4H    0020H  PUB  INITIALISATION
01B4H    011DH  PUB  INIT_TRANSF_MATR          01B4H    004BH  PUB  RANGE
                     -IX
01D2H    01C0H  PUB  PQCALL_DEVICEDRI          01D2H    0168H  PUB  PQINTERNALWRITEI
                     -VER                                         -UFFER
01D2H    00B0H  PUB  PQUTSCLOSEDOWN            01D2H    0060H  PUB  PQUTSSTARTUP
01D2H    0160H  PUB  PQ_FAR_INTERNAL_          01D2H    0100H  PUB  PQ_GET_LOCAL_TOK
                     -WRITE_BUFFER                                -EN
01D2H    0120H  PUB  PQ_IOSC                   01D2H    0110H  PUB  PQ_STORE_LOCAL_T
                                                                  -OKEN
01D2H    0220H  PUB  TQALLOCATE                01D2H    0210H  PUB  TQEXIT
01D2H    0200H  PUB  TQGETERH                  01D2H    01D0H  PUB  TQINITIALIZE
01D2H    0260H  PUB  TQPARSECL                 01D2H    0000H  PUB  TQ_001
01D2H    0140H  PUB  TQ_302                    01D2H    0020H  PUB  TQ_999
01D2H    0040H  PUB  TQ_COPYRIGHT              0258H    0000H  PUB  TQLOCAL
0258H    0004H  PUB  TQACCESS                  0258H    0002H  PUB  TQCURRENTPTR
0258H    0005H  PUB  TQDATAAREAFREE            01D2H    0290H  PUB  TQDEFAULTPL
01FCH    0000H  PUB  INITFP                    01FDH    003FH  PUB  CI
01FDH    0044H  PUB  CO                        01FDH    0005H  PUB  DQCALLI
01FDH    000AH  PUB  DQCFG                     01FDH    000FH  PUB  DQFAIL
01FDH    0000H  PUB  DQSEND                    0367H    0039H  PUB  MBRI
0367H    0058H  PUB  MBRLI                     0367H    0000H  PUB  MBWI
0367H    0019H  PUB  MBWLI
```

MEMORY MAP OF MODULE PRINCIPLE

MODULE START ADDRESS PARAGRAPH = 0100H OFFSET = 0006H
SEGMENT MAP

START	STOP	LENGTH	ALIGN	NAME	CLASS	OVERLAY
01000H	01018H	0019H	A	??LOC86_INITCO-DE	CODE	
01400H	01A8DH	068EH	W	PRINCIPLE_CODE	CODE	
01A8EH	01B41H	00B4H	W	CONSOLE_CODE	CODE	
01B42H	01D18H	01D7H	W	HARDWARE_CODE	CODE	
01D20H	01FB4H	0295H	G	UTSCODE	CODE	
01FC0H	01FC0H	0001H	G	LIB_87_NULLP	CODE	
01FD0H	02018H	0049H	G	CODE	CODE	
02400H	0257FH	0180H	W	PRINCIPLE_DATA	DATA	
02580H	02580H	0000H	W	CONSOLE_DATA	DATA	
02580H	02580H	0000H	W	HARDWARE_DATA	DATA	
02580H	02581H	0002H	G	UTS_DATA_SEGME-NT	DATA	
02582H	02585H	0004H	B	TQNULLDEVICE	DATA	
02586H	02586H	0001H	B	TQNULLDATAAREA	DATA	
02590H	02593H	0004H	G	PCLIST_BASE	DATA	
03400H	03669H	026AH	G	STACK	STACK	
03670H	03670H	0000H	G	??SEG		
03670H	036F2H	0083H	G	CODE		
03700H	03700H	0000H	G	MEMORY	MEMORY	

GROUP MAP

ADDRESS	GROUP OR SEGMENT NAME
01D20H	UGROUP
	UTSCODE

Appendix XVI

REFERENCES

Former Reports:

E. Galet : Teach operations with a force sensor for a linear robot arm - February 1986 - WPB report n°0255.

L. Janvier : Teaching operations with a force sensor for a linear robot arm - February-june 1986 - WPA report n°0306.

E. Retrain : Control Unit for a two dimentional robot - September 87-February 88 - WPA report n°0542.

Intel Books:

- MCS-86 User's Manual (ref: 9800722A).
- iSBC 86/12 Single Board Computer Hardware Reference Manual. (ref: 9800645A).
- Peripheral Design Handbook.
- PASCAL 86 User's Guide (ref: 121539-005).
- iSDM86 system Debug Monitor (ref:146165-001).
- IAPX 86-88 Family Utilities User's Guide (ref: 121616-004).

Scientific Books:

Ralph MOrrison : Instrumentation Fundamentals and Applications - John Willey & Sons 1984 -.

Grounding and Shelding Techniques in Instrumentation - John Willey & Sons 1986-.

Asada H., Slotine J.J.E. : Robot Analysis and Control - Whiley Interscience Publication 1986 -.

Data Books:

- Burr-Brown : Integrated Circuits Data Book.
- Analog-Devices : Linear Product Data Book.
- Texas-Instrument : Integrated Circuits Data book.